

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МАТЕМАТИЧЕСКОЙ ТЕОРИИ МИКРОПРОЦЕССОРНЫХ
СИСТЕМ УПРАВЛЕНИЯ

Леонов Даниил Олегович

Выпускная квалификационная работа бакалавра

Создание сервисного приложения

Направление 010400

Прикладная математика и информатика

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Петрова В. А.

Санкт-Петербург

2018

Оглавление

Введение	3
Обзор существующих решений	4
Постановка задачи	13
Обзор литературы	15
Глава 1. Анализ предметной области	16
Глава 2. Разработка сервисного приложения	21
2.1. Анализ языков программирования	21
2.2. Анализ фреймворков языка Python	24
2.3. Разработка базы данных для отдельного магазина	27
2.4. Описание функционала пользователей	33
Глава 3. Особенности реализации приложения с помощью средств Django ..	39
3.1. Описание основных алгоритмов реализации программного продукта	39
3.2. Описание разработанных статусов заказа	47
3.3. Описание процесса создания интернет-магазина	48
3.4. Описание работы пользователей в созданном интернет-магазине	50
3.5. Тестирование разработанного продукта	64
Выводы	67
Заключение	68
Список цитируемой литературы	70
Приложение	73

Введение

Многим компаниям для увеличения прибыли необходимо выходить на новые рынки продаж. Одним из таких рынков для отдельной компании могут стать продажи через Интернет. Но для выхода на такой рынок необходимы программные решения, позволяющие это сделать. Часто бывает так, что такие программные решения или слишком дорогостоящие, или сложны и требуют специальный постоянный штат сотрудников, способный настраивать и поддерживать стабильную работу таких систем. В связи с этим, актуальной задачей является разработка удобной системы для создания и поддержки интернет-магазина с простым и понятным пользовательским интерфейсом.

В данной работе представлено подобное сервисное приложение с описанием его разработки и реализации. Конкурентными преимуществами данного приложения должны стать: простота использования и настройки, удобный интерфейс, стабильность и универсальность, также приложение должно быть свободно распространяемым. Более того, разработанное сервисное приложение должно подходить для многих компаний, с разными объемами продаж, численностью сотрудников и количеством торговых точек. Пользователями разработанного приложения могут быть, как и индивидуальный предприниматель, стремящийся систематизировать своих клиентов и товар, так и крупная оптовая компания, имеющая много торговых точек по всему городу. Помимо этого, пользователями создаваемой системы будут покупатели – клиенты компании.

Обзор существующих решений

Для того, чтобы создать конкурентный программный продукт в первую очередь необходимо проанализировать существующие решения. В данном обзоре рассматриваются приложения, позволяющие автоматизировать процесс создания интернет-магазина, а также позволяющие создавать систему управления заказами, товарами, клиентами, сотрудниками и торговыми точками компании.

1. Конструктор WIX.COM

Первым решением рассматривается конструктор сайтов и различных интернет-магазинов wix.com. Личный кабинет в конструкторе представлен на рис. 1. Это один из самых популярных конструкторов, который работает с 2006 года. У данного конструктора имеется много различных шаблонов с красивым дизайном. Визуально рабочая область выглядит удобно и понятно.

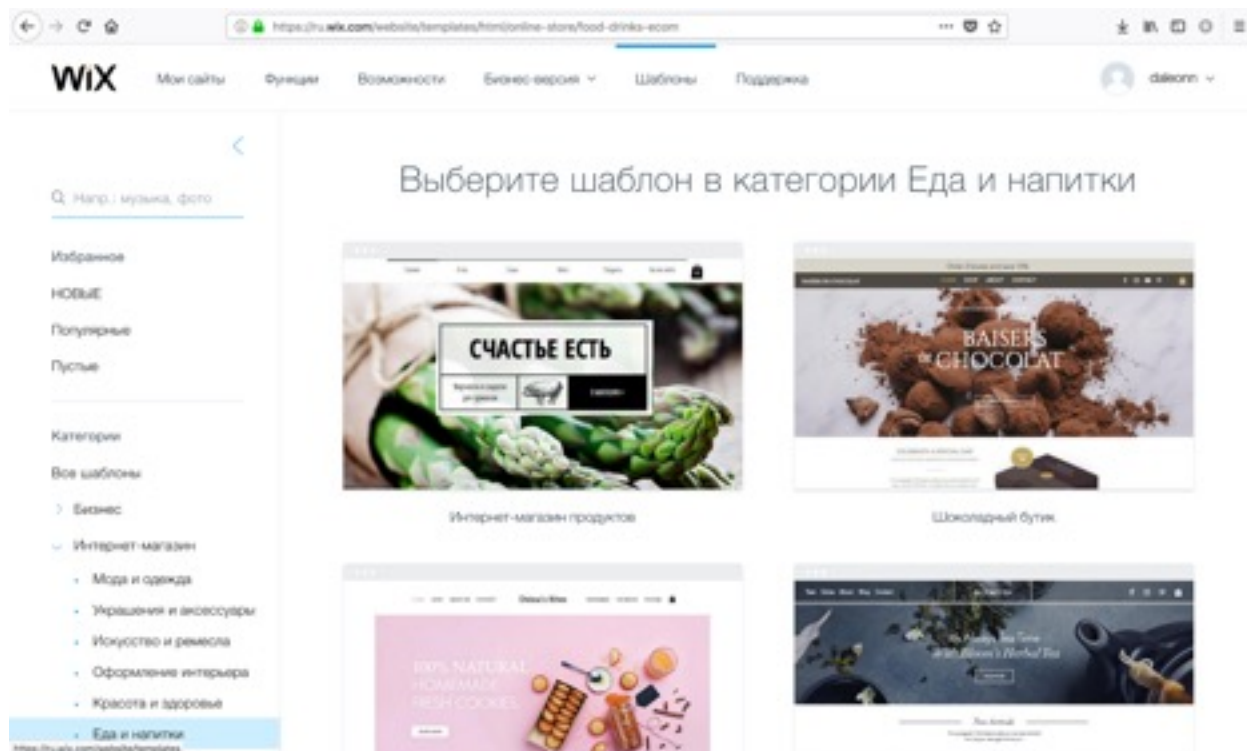


Рис. 1 – Конструктор сайтов WIX

Чаще всего с помощью данного конструктора создают простые и красивые сайты-визитки.

Основными минусами этого приложения можно выделить следующие моменты:

- из-за того, что конструктор имеет множество шаблонов и встроенных функций, приложение не всегда работает быстро, особенно на не мощных компьютерах с медленным Интернет-соединением;
- громоздкий код созданного интернет-магазина, а также отсутствие возможности его правки. Правда, это может быть и плюсом, потому что неопытные пользователи могут изменить код таким образом, что это приведет к неустранимым ошибкам;
- неудобный и некрасивый домен, на котором создается интернет-магазин. Все созданные с помощью данного конструктора сайты имеют бесплатный домен следующего вида: *.wix.com/название ресурса, где * – логин пользователя, создавшего сайт;
- несмотря на простоту в «сборке» интернет-магазинов, необходимо потратить много времени на его комплектацию, а также на создание форм регистрации и внутренней системы заказов;
- созданные интернет-магазины имеют отметку о том, что сайт создан с помощью данного конструктора, убрать которую можно только приобретя один из тарифных планов.

2. Конструктор NETHOUSE

Следующим инструментом рассматривается один из лучших конструкторов интернет-магазинов nethouse.ru. Личный кабинет конструктора представлен на рис. 2. Данный конструктор легок в освоении, а также имеет около сотни различных шаблонов. Одним из плюсов данного конструктора является возможность подключения сторонних систем учета товаров и заказов.

Минусом данной системы можно выделить сложность добавления в систему и настройки взаимодействия сотрудников и торговых точек компании. Данный конструктор является одним из лучших в обзоре.

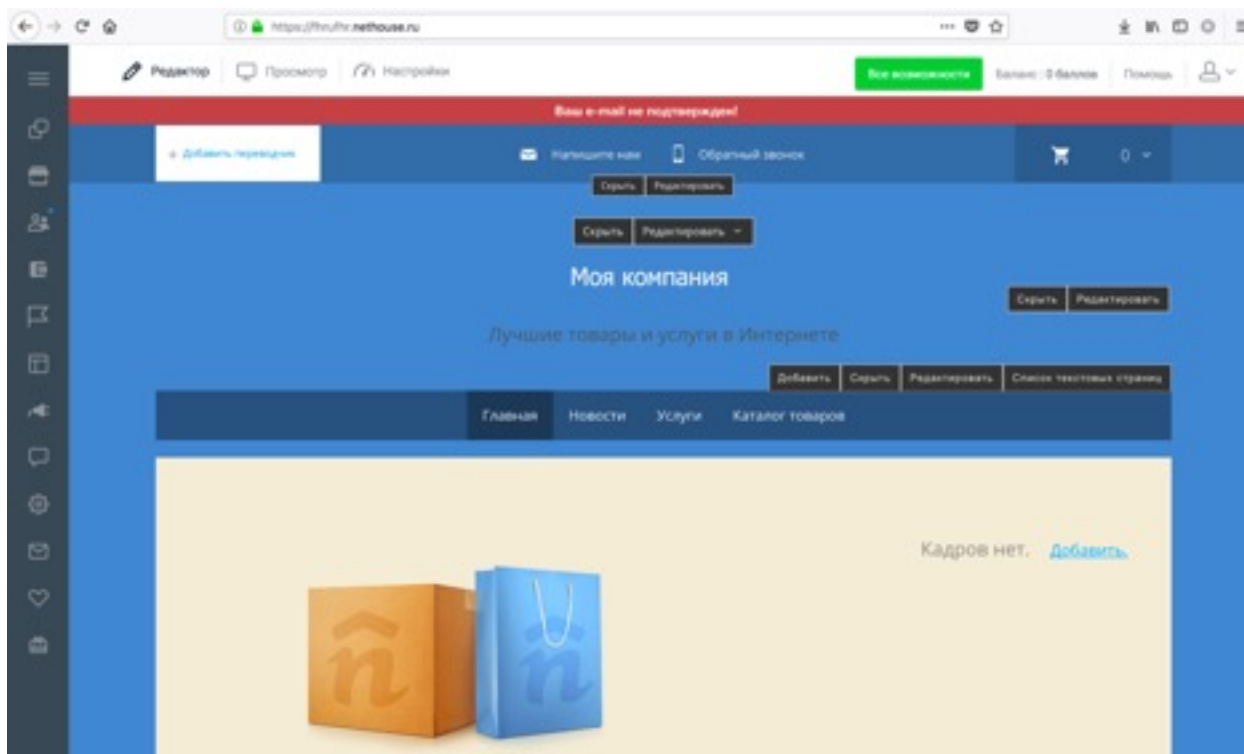


Рис. 2 – Конструктор сайтов nethouse

3. Конструктор Ukit

Приложение для создания интернет-магазинов Ukit является новым на рынке. Оно популярно среди предпринимателей, владельцев малого и среднего бизнесов. Личный кабинет конструктора представлен на рис. 3. Это достаточно универсальный конструктор, с помощью которого можно сделать сайт-визитку, блог, форум, а также необходимый интернет-магазин. У данного конструктора мощный функционал, но для его использования требуется высокий профессиональный уровень. В качестве плюса в создании интернет-магазина на данной платформе можно выделить удобный домен 2-го уровня, который предоставляется бесплатно.

Основными недостатками данного приложения можно выделить следующее:

- долгий процесс регистрации и создания интернет-магазина, необходимо пройти множество этапов подтверждения личных данных перед непосредственной настройкой и комплектацией интернет-магазина;
- сложный интерфейс, на ознакомление с которым новичкам потребуется время;
- необходимо большое количество времени на комплектацию магазина и настройку системы управления заказами, сотрудниками и торговыми точками.

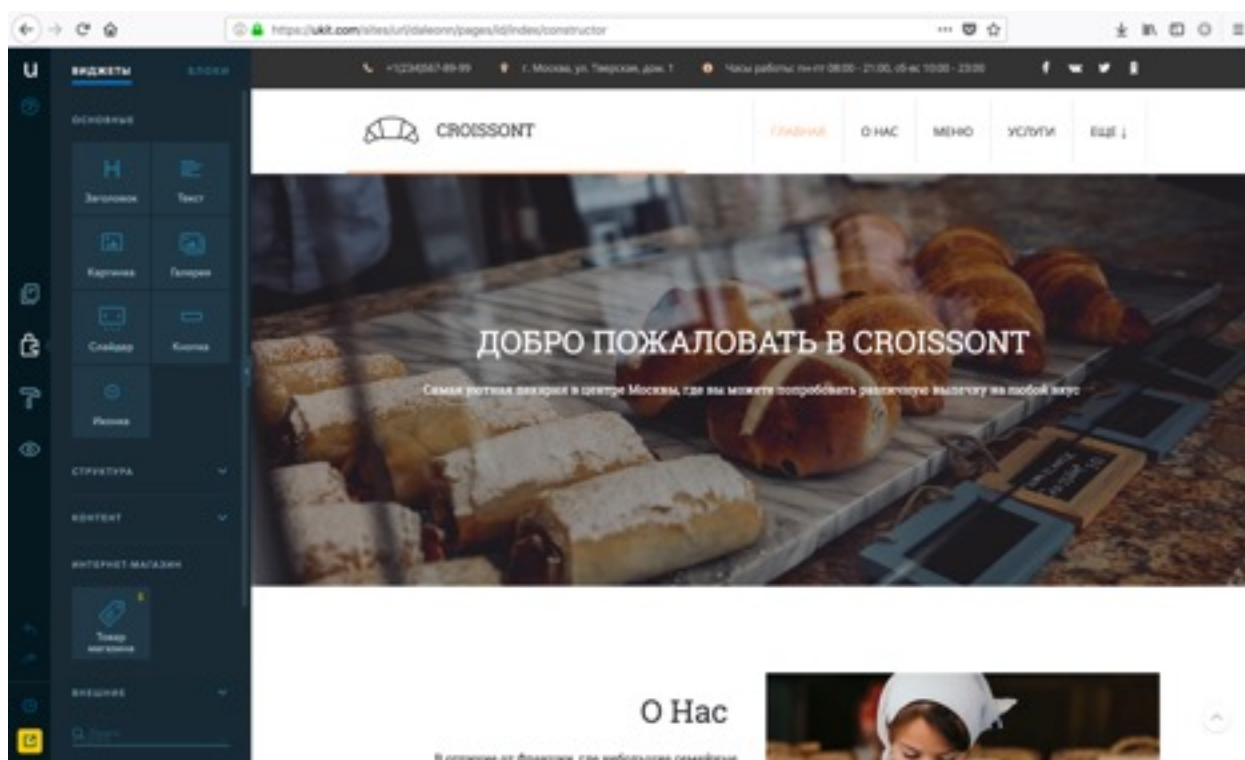


Рис. 3 – Конструктор сайтов UkIt

4. Конструктор Umi

Следующим рассматриваемым приложением является Umi. Это простая и легкая в освоении система, обладающая большим функционалом, в том числе привязкой к системам по управлению складом и 1С. Конструктор

имеет много готовых шаблонов, которые не очень сложно редактировать. Помимо этого, система достаточно популярна, что подтверждается большим количеством сайтов, созданных с помощью данной платформы (около 1,5 млн.). На рис. 4 представлена панель администратора сайта.

Минусами данной системы можно выделить следующее:

- практически за весь функционал необходимо платить;
- длительное время, необходимое для создания магазина, а также подключения сторонних систем;
- долгий процесс регистрации администратора будущего сайта в системе;
- стоимость минимальной оплаты для работы сайта составляет 550 рублей в месяц.

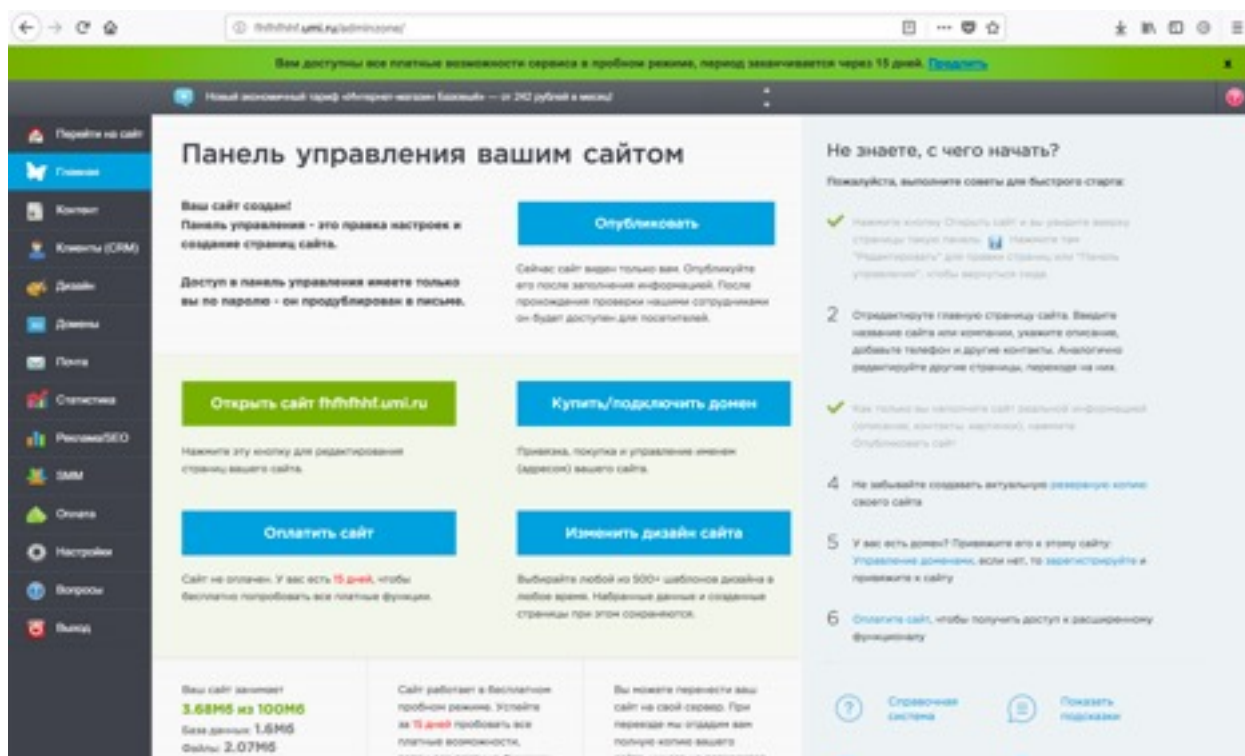


Рис. 4 – Конструктор сайтов Umi

5. Конструктор a5.ru

Конструктор a5.ru является российский проектом, который создан в 2011 году. У пользователей данного приложения, как и у всех других приложений, есть возможность создания сайтов-визиток, одностраничных сайтов и интернет-магазинов. У конструктора есть около 300 шаблонов, но большинство из них не являются адаптивными. Помимо этого, отсутствуют шаблоны для создания интернет-магазинов, точнее они есть, но представляют собой каталоги, без возможности оформления заказа. Личный кабинет администратора представлен на рис. 5.

Таким образом, данное приложение имеет следующие недостатки:

- неадаптивные шаблоны;
- отсутствие шаблонов для создания интернет-магазина, и соответственно, отсутствие системы для контроля заказов, клиентов, торговых точек;
- медленная работа приложения.

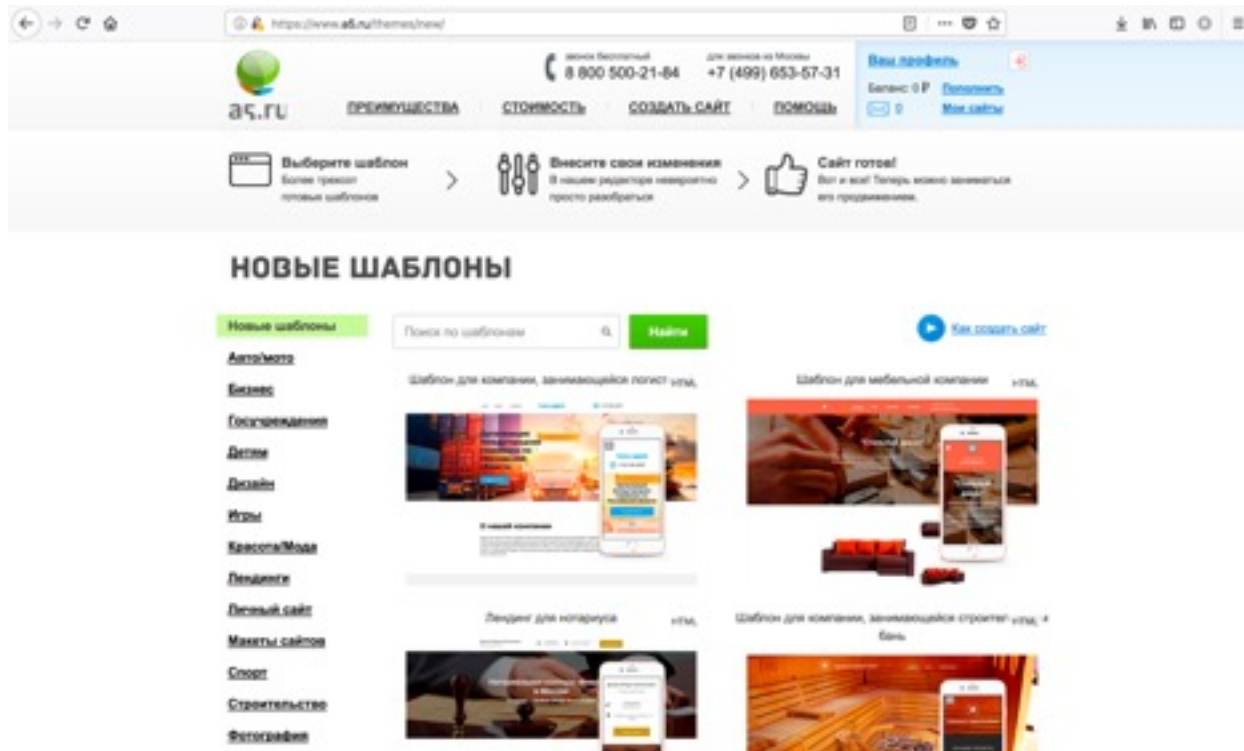


Рис. 5 – Конструктор сайтов a5.ru

6. Конструктор Tilda

Последним рассматриваемым конструктором является Tilda, личный кабинет в котором представлен на рис. 6. С Tilda можно создавать сайты компании, различные визитки и небольшие интернет-магазины. Все действия происходят во внутреннем визуальном редакторе, но не для профессионала будет сложно освоиться с созданием блоков и настройки их взаимодействия.

Главным плюсом Tilda является дизайн разработанных сайтов, потому что все шаблоны разработаны профессиональными дизайнерами.

Минусами данной системы является следующее:

- сложность создания интернет-магазина и системы управления заказами, товарами и сотрудниками;
- высокая стоимость тарифного плана, который необходим в связи с тем, что базовый позволяет создавать лишь несколько страниц;
- нацеленность приложения на создание сайтов-визиток;
- большое количество времени, необходимого для овладения данной системой, а также для создания интернет-магазина.

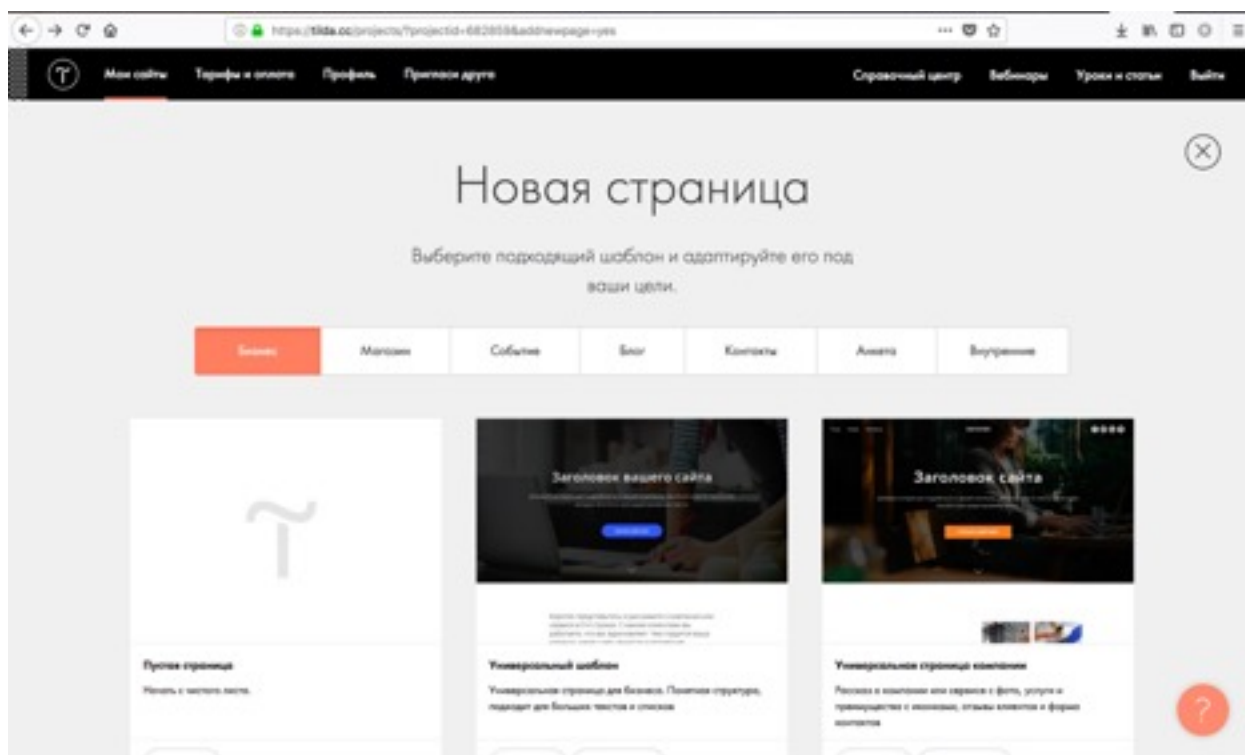


Рис. 6 – Конструктор сайтов Tilda

Ниже представлена таблица 1, содержащая результаты сравнительного анализа приложений-конструкторов. Во время их сравнения для всех параметров, кроме «Количество этапов, которое необходимо пройти при регистрации нового магазина», введены условные оценки – лингвистические переменные, которые градуируются на числовой оси [1]. Так, числу 1 соответствует переменная «очень плохо», 2 – «плохо», 3 – «нормально», 4 – «хорошо», 5 – «отлично». Оценка * в параметре «Наличие внутренней системы для поддержки заказов, пользователей и товаров» означает, что такая система отсутствует, но может быть интегрирована сторонняя.

Параметр	Рассматриваемое приложение					
	WIX	Nethouse	Ukit	Umi	a5	Tilda
Скорость работы приложения	2	5	5	4	4	4
Количество этапов, которое необходимо пройти при регистрации нового магазина	3	2	4	4	2	3
Выбор бесплатного домена	3	5	5	5	3	5
Конечная цена	3	4	4	3	3	2
Количество и качество шаблонов для создания интернет-магазинов	5	5	5	5	2	5
Наличие внутренней системы для поддержки заказов, пользователей и товаров	1	*	1	*	1	1
Простота и скорость создания элементарного интернет-магазина	4	5	5	5	3	3
Простота и скорость настройки созданного интернет-магазина	4	5	4	4	3	3

Таблица 1 – Сравнение существующих приложений

Проведя анализ существующих программных продуктов, можно выдвинуть ряд требований к созданию конкурентоспособного приложения.

Приложение должно обеспечивать выполнение следующих условий:

- простота и скорость создания нового интернет-магазина, то есть он должен создаваться максимально быстро по задаваемому адресу;
- минимальное время для регистрации в системе. Пользователь не должен тратить много времени на верификацию своих данных, к тому же не все пользователи хотят делиться своими личными данными, такими как, например, телефон;
- наличие удобного, универсального, современного, интуитивно понятного и адаптивного шаблона интернет-магазина;
- иметь автоматически создаваемые формы регистрации для клиентов и сотрудников, а также личные кабинеты. Пользователь не должен разбираться в методах создания авторизации и тратить на это время;
- создаваемые в системе магазины должны располагаться на доменах второго уровня, это необходимо для красоты и краткости адреса, на котором будут располагаться магазины. К тому же, пользователь должен иметь возможность выбора этого адреса, для придания узнаваемости созданному магазину;
- конечная цена для пользователя должна быть максимально низкой;
- наличие интуитивно понятного интерфейса для работы с системой заказов, товаров, торговых точек и сотрудников;
- наличие удобной системы выгрузки товаров в систему, например, одним файлом, в котором содержится вся необходимая информация.

Постановка задачи

Целью данной работы является создание сервисного приложения для автоматизации процесса создания интернет-магазина компании в сфере продаж, а также поддержка управления ассортиментом товаров, списка заказов, перечнем клиентов, торговых точек и сотрудников компании.

Для достижения поставленной цели необходимо последовательно решить следующие задачи:

- проанализировать работу компании, которая занимается продажами;
- провести анализ существующих средств разработки веб-приложений и осуществить выбор тех, которые являются лучшими для достижения поставленной цели на данный момент;
- разработать и описать универсальную базу данных товаров, заказов, торговых точек и клиентов для компаний, использующих сервисное приложение;
- разработать и описать функционал всех возможных пользователей в системе;
- провести описание всех алгоритмов, используемых в ходе разработки программного продукта;
- провести тестирование созданного программного решения;
- апробировать сервисное приложение в условиях реальной компании.

Функционал разрабатываемого программного продукта

С помощью созданного сервисного приложения клиент сможет:

- просматривать электронный каталог товаров компании, а также ее торговые точки;
- создавать заказы, состоящие из необходимых товаров, указывать адрес доставки;
- контролировать этапы выполнения заказа в личном кабинете.

Сотрудники компании, в зависимости от их статуса, смогут:

- водитель-курьер: просматривать необходимые для доставки заказы и менять их статус;
- работник магазина или склада: просматривать необходимые заказы, а также товары магазина, к которому привязан сотрудник;
- менеджер магазина или склада: просматривать все заказы, товары и сведения о сотрудниках магазина, добавлять туда товары из каталога, а также создавать, удалять и изменять заказы;
- менеджер компании: просматривать все заказы, товары, а также сведения о сотрудниках и клиентах компании. Добавлять товары в общий каталог товаров и распределять их по магазинам;
- администратор: создавать интернет-магазин и всячески его настраивать: добавлять и редактировать магазины, пользователей системы. Помимо этого, администратор будет иметь все возможности других сотрудников компании.

Обзор литературы

При выполнении данной работы использована различная современная, а также классическая литература, необходимая для проектирования и разработки веб-приложений.

Для введения и описания условных оценок при сравнении существующих программных решений использовалась классическая книга по нечеткой математике А. Пегата [1]. Для описания и структурирования информационных потоков в торговой компании была использована книга А. Тысленко [2].

Во время выбора программных инструментов для разработки была использована литература, описывающая и сравнивающая все анализируемые инструменты [5–14]. Некоторые из этих используемых источников являются англоязычными.

Перед тем, как начать разработку сложного веб-приложения, необходимо правильно определить этапы разработки [3, 4]. В ходе разработки веб-приложения основным языком программирования был Python, теоретические и практические основы которого описаны в работе Лутца М. [9], также использовались источники по фреймворку Django [11, 13] и литература по остальным необходимым в ходе разработки инструментам [15, 16, 17, 18]. Универсальная база данных для торговой компании была разработана с использованием [19, 21, 22].

Источники [24–26] необходимы для построения рейтингов популярности браузеров и систем управления базами данных.

Глава 1. Анализ предметной области

Объектом анализа в данной работе являются компании, занимающиеся продажами. Данные компании уже полностью или частично осуществляют свою деятельность через Интернет, или же только хотят выйти на новый для себя рынок интернет-продаж.

Компании в сфере продаж могут быть любыми. Они могут отличаться как по объему продаж, ассортименту товаров, количеству сотрудников, клиентов и торговых точек, так и по виду продаваемого товара. На рис. 7 представлена базовая структурная схема информационных потоков в торговой компании [2]. Базовой она является из-за того, что каждая компания уникальна, и у нее могут быть какие-то иные, не представленные в схеме подразделения, а также некоторые подразделения могут и вовсе отсутствовать. В прямоугольных блоках показаны подразделения компании, стрелками указан поток информации между ними. Единственная стрелка, соединяющая подразделения «Заказы» и «Топ-менеджеры» выделена пунктиром, потому что это внешний поток информации.

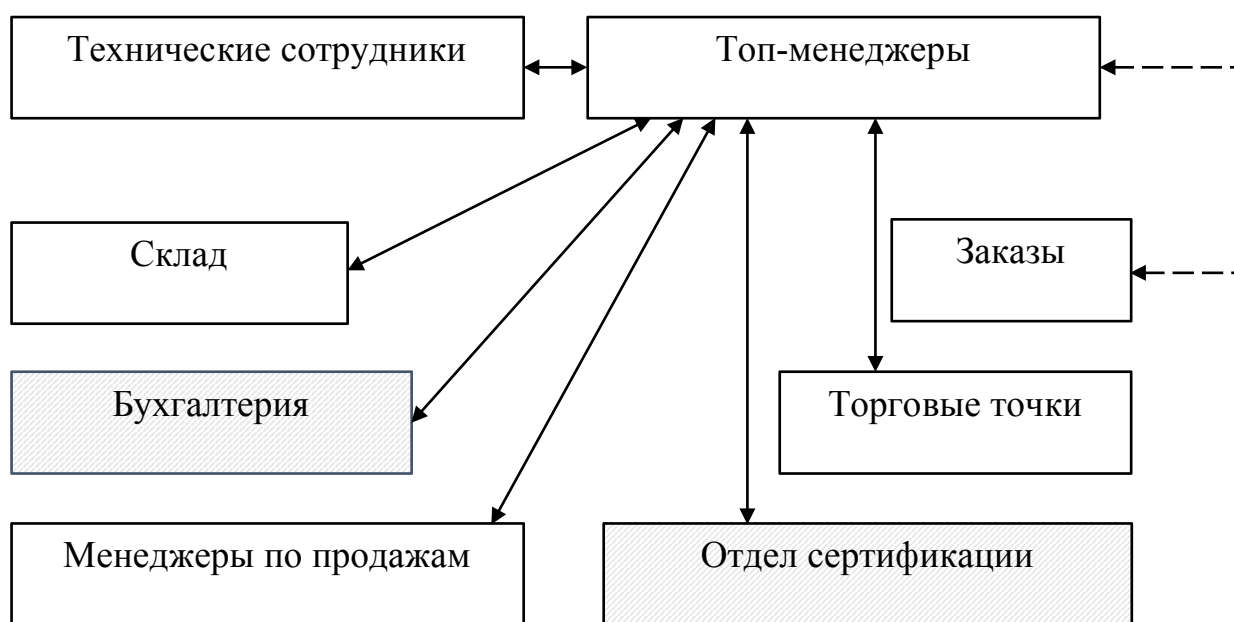


Рис. 7 – Базовая структурная схема информационных потоков в торговой компании

Далее рассматриваются следующие подразделения: «Технические сотрудники», «Склад», «Торговые точки», «Менеджеры по продажам», «Топ-менеджеры» и «Заказы». Подразделения «Бухгалтерия» и «Отдел сертификации» рассматриваться не будут, так как у них особые задачи, а данная работа нацелена только на автоматизацию процессов создания и выполнения заказов, поддержки торговых точек и ассортимента товаров.

1.1. Подразделение «Склад»

В данной работе подразумевается, что компания, которая будет пользоваться разработанным продуктом, имеет физический склад. На самом деле, компания может и не иметь его как отдельный физический объект, тогда предполагается, что склад – это просто совокупность всех товаров, которые в данный момент находятся в состоянии продажи или сборки для отправки. Это уточнение необходимо для того, чтобы веб-приложением смогли пользоваться не только крупные компании, но и индивидуальные предприниматели, которые не имеют склада. На складе хранится товар, а также к складу привязаны сотрудники, которые на нем работают. Это могут быть грузчики-комплектовщики и менеджеры склада. Помимо этого, товар необходимо доставлять клиентам. В данной разрабатываемой системе будет подразумеваться, что компания может иметь или собственных водителей-курьеров, или привлекать сторонних, но тогда, сторонним сотрудникам необходимо будет также зарегистрироваться в системе для получения информации о заказах, адресах доставки и отслеживания менеджерами сохранности доставки. На склад может поступать товар от поставщика, или появляться по мере его производства, таким образом, менеджер склада может добавлять поступающий товар в систему. Основная задача подразделения «Склад» заключается в комплектации склада товарами, комплектации заказов и их отправки клиентам.

1.2. Подразделение «Торговые точки»

У компании, занимающийся продажами, могут быть собственные торговые точки или магазины, в которых они реализуют товар. У каждой такой торговой точки есть свои характеристики, такие как адрес, режим работы и товар, который продается именно в этой торговой точке. В каждом магазине компании работают сотрудники компании. Они делятся на работников и на менеджеров магазина. Первые работают непосредственно в торговом зале, собирая товар в заказах и расставляя его по полкам, а вторые управляют своей торговой точкой и сотрудниками, а также занимаются реализацией товара. Подразумевается, что клиент-покупатель может заказать товар прямо в конкретном магазине и заехать забрать его без очереди по выгодной цене. Существуют компании, у которых может и не быть таких торговых точек, то есть они занимаются только хранением товара на складе и отправкой его другим клиентам. Основной задачей подразделения «Торговые точки» является реализация товаров и их прием, а также сборка заказов и передача их клиентам, которые хотят забрать свой заказ самовывозом.

1.3. Подразделение «Менеджеры по продажам»

В компании работают менеджеры, которые занимаются продажами, а также закупкой необходимых товаров у поставщиков. Данные менеджеры занимаются работой с клиентами – юридическими и физическими лицами, которые имеют возможность заказывать товар с доставкой по определенному адресу. Менеджерам по продажам доступна информация о поставщиках, для связи с ними. Основной задачей подразделения «Менеджеры по продажам» является контроль выполнения заказов, а также выбор поставщиков для заказа товаров.

1.4. Подразделение «Топ-менеджеры»

К данному подразделению относятся топ-менеджеры компании, которые управляют всеми подразделениями компании и имеют полный доступ к информации о компании. Основная задача этого подразделения – поддержание взаимодействия и корректности работы других подразделений.

1.5. Подразделение «Технические сотрудники»

Сюда можно отнести технический отдел, который отвечает за взаимодействие всех подразделений компании. Основная задача данного подразделения – комплектация интернет-магазина всей необходимой информацией и поддержка корректности его работы.

1.6. Подразделение «Заказы»

К данному подразделению относятся все заказы, поступающие в магазин. Данное подразделение включает в себя внешний поток информации, так как заказы появляются в системе только после того как их создает клиент-покупатель компании.

Ниже приведена таблица 2, в которой описано распределение задач между сотрудниками подразделений компании.

		Выполняемая задача					
Подразделение	Сотрудник	Сборка и контроль заказов своей точки	Доставка заказов	Поддержка товаров, заказов и сотрудников своей точки	Поддержка взаимодействия сотрудников	Поддержка заказов, взаимодействие с поставщиками и клиентами	Комплектация магазинов товарами, сотрудниками
Склад	Комплектовщик	V					
Склад	Водитель		V				
Склад	Менеджер			V			
Торговая точка	Работник магазина	V					
Торговая точка	Менеджер			V			
Менеджер по продажам				V		V	
Топ-менеджеры				V		V	V
Технический сотрудник					V		V

Таблица 2 – Распределение задач между сотрудниками подразделений компании

Глава 2. Разработка сервисного приложения

Для того чтобы разработать корректно и надежно функционирующее сервисное веб-приложение, а в дальнейшем поддерживать его стабильную работу необходимо проанализировать и выбрать правильные инструменты [3, 4, 21].

2.1. Анализ языков программирования

Основным инструментом разработки веб-приложения является язык программирования, имеющий удобные средства для разработки и реализации способов хранения необходимой информации об интернет-магазине на сервере и реализации клиент-серверного обмена с пользователями. Его необходимо выбрать в первую очередь, так как от этого будет зависеть выбор дальнейших средств разработки. В процессе выбора языка программирования необходимо провести анализ существующих языков программирования. Ниже рассмотрены языки программирования, которые чаще всего используются при создании веб-приложений, такие как Ruby, Python и PHP, и проводится их сравнение.

PHP (hypertext preprocessor) – скриптовый язык программирования общего назначения. PHP используется для написания сценариев, которые будут выполняться на стороне сервера. Данный язык программирования может выполнять различные операции с HTML-документами, перенаправлять на другие страницы веб-приложения, управлять аутентификацией пользователей, обрабатывать файлы, загружаемые на сервер, и предоставлять доступ к базе данных [5, 6, 7].

Основные преимущества РНР:

- достаточно легок в освоении;
- свободно распространяемый;
- очень популярен, поддерживается большим сообществом программистов;
- включает в себя поддержку основных возможностей объектно-ориентированного программирования;
- имеет большое количество расширений и библиотек.

Основные недостатки РНР:

- невозможность асинхронной и многопоточной работы приложения;
- низкое качество кода;
- несогласованность синтаксиса функций;
- веб-приложения, которые написаны на РНР, часто имеют проблемы с безопасностью.

Ruby – объектно-ориентированный язык программирования высокого уровня. Язык обладает независимой от операционной системы реализацией многопоточности, а также строгой динамической типизацией. С помощью Ruby можно быстро и просто программировать, так как нет необходимости объявлять переменные и их типизировать. Есть хорошая поддержка операций со строками и регулярными выражениями. Существует очень мощный фреймворк Ruby-On-Rails для создания веб-приложений [7, 8].

Основные преимущества Ruby:

- возможность расширения возможностей языка с помощью множества библиотек;
- интегрируемость с различными СУБД, такими как: MySQL, DB2, Oracle;
- простой и удобный интерфейс;

- имеет средства для работы с массивами и строками;
- наличие фреймворков для создания веб-приложений.

Основные недостатки Ruby:

- сложность в освоении;
- отсутствие необходимого количества документации;
- менее производителен, в сравнении с Python и PHP.

Python, как и Ruby, является объектно-ориентированным языком программирования высокого уровня. Синтаксис этого языка минималистичен и прост, стандартные библиотеки имеют очень большое количество полезных функций [7]. Python используется для решения различных задач, в том числе для создания веб-приложений. С помощью Python можно динамически генерировать HTML-код со стороны сервера [9].

Основные преимущества Python:

- удобный и легко понятный программный код;
- удобное и быстрое создание моделей системы;
- большое количество документации, а также быстрорастущее сообщество программистов, использующий Python;
- объектно-ориентированный язык;
- огромное количество различных библиотек;
- разнообразие фреймворков для создания веб приложений.

Основные недостатки Python:

- не самый популярный язык программирования;
- несовершенство многопоточности;
- начальная ограниченность средств работы с базой данных, но впоследствии устранимая.

В таблице 3 представлено сравнение языков Ruby, PHP и Python по другим параметрам, некоторые из которых влияют на процесс разработки продукта.

Параметр	Python	PHP	Ruby
Год релиза	1991	1995	1995
Сложность изучения	Самый простой	Сложнее Python	Самый сложный
Объектно-ориентированность	Да	Да	Да
Автоматический сборщик «мусора»	Да	Да	Да
Возможность компиляции	Да	Да	Да
Множественное наследование	Да	Нет	Нет
Наличие различных библиотек	Да	Да	Да
Многомерные массивы	Да	Да	Да
Наличие различных фреймворков	Да, больше чем у конкурентов	Да	Да
Многомерные массивы	Да	Да	Да
Удобство поддержки кода	Самый удобный	Самый неудобный	Удобнее PHP

Таблица 3 – Сравнение Ruby, PHP и Python

Исходя из сравнения языков программирования было принято решение использовать язык программирования Python последней версии 3.6. Этот язык наиболее прост в изучении, а также обладает хорошей интеграцией с СУБД.

2.2. Анализ фреймворков языка Python

После того как языком программирования был выбран Python версии 3.6 необходимо выбрать фреймворк для создания веб-приложений. Фреймворк необходим для того, чтобы, по сути, пользователи разработанного приложения могли работать с разработанным приложением через браузер. Фреймворк облегчает разработку программного продукта и объединяет разные компоненты программной системы.

Для выбора фреймворка необходимо проанализировать существующие средства разработки веб-приложений на языке Python. Существует несколько решений: Django, Flask, Pyramid и web2py. Однако, Django и Flask являются наиболее популярными среди разработчиков, поэтому анализировать и сравнивать необходимо именно эти фреймворки.

Flask создан австрийским разработчиком Армином Ронахером в 2010 году. Django – фреймворк, который разработан Django Software Foundation и выпущен в 2005 году. Стоит отметить, что оба фреймворка являются бесплатными с открытым программным кодом [12, 13].

Необходимо выделить основные различия между Flask и Django [10, 11, 12, 13]. Во-первых, Django чаще всего используют при создании сложных веб-приложений, в то время как Flask предназначен для небольших веб-приложений с одной-двумя функциями.

Во-вторых, при знакомстве с новыми средствами разработки необходимо обратить внимание на их документацию. Документация Django намного подробнее и ее больше, чем документация Flask. Это говорит о том, что будет процесс изучения Django будет понятнее и легче.

В-третьих, Django имеет встроенную ORM – виртуальную объектную базу данных, которая позволяет работать с различными видами баз данных. Flask имеет возможность подключения сторонней ORM, например, SQLAlchemy, но всегда удобнее использовать уже готовое встроенное решение.

В-четвертых, при создании приложения с помощью Django автоматически создается строго структурированная система директорий. Используя Flask, разработчик сам создает структуру своего приложения. Может показаться, что в данном случае удобнее и проще использовать Flask, но это не так. В случае создания сложного приложения с множеством функций с помощью Flask гораздо труднее структурировать код в целом при отсутствии специальных средств для этой цели.

Одним из самых мощных преимуществ Django является автоматически создаваемый интерфейс администратора базы данных. С помощью данного интерфейса доверенные пользователи могут удобно управлять базой данных веб приложения.

Из сравнения Django и Flask видно, что намного лучше для решения задачи подходит фреймворк Django, поэтому он и будет использоваться при разработке веб-приложения.

Следующим этапом в выборе инструментов разработки является выбор системы управления базы данных [14]. По умолчанию фреймворк Django использует СУБД SQLite. Когда приложение использует SQLite, то интеграция работает с функциональными и прямыми вызовами, сделанными в файл, содержащий данные, вместо того, чтобы обмениваться данными через интерфейс. В результате этого облегчается нагрузка на сервер, в сравнении с другими СУБД. Это делает SQLite чрезвычайно быстрым и эффективным, а также мощным, благодаря базовой технологии библиотеки. Помимо этого, SQLite входит в десятку самых популярных СУБД [23].

Таким образом, опираясь на проведенный анализ, для разработки сервисного приложения были выбраны следующие технологии: язык программирования Python 3.6, фреймворк языка Python Django, HTML, CSS, JavaScript, СУБД SQLite 3, веб-сервер Apache 2.0 [15, 16, 17, 18]. Помимо перечисленных инструментов, необходим виртуальный приватный сервер, на котором будет располагаться исходный код разрабатываемого приложения. VPS куплен у компании Digital Ocean, а также приобретено доменное имя daleonn.store, на котором будет расположена форма создания интернет-магазина, а на его поддоменах будут находиться адреса создаваемых магазинов.

2.3. Разработка базы данных для отдельного магазина

Для разработки базы данных необходимо построить модель «сущность – связь» [19].

В ходе разработки универсальной базы данных для торговой компании было выделено 11 сущностей, используемых приложением [19, 20, 21, 22]:

- поставщик;
- товар;
- товар в заказе;
- товар в магазине;
- магазин (склад);
- сотрудник;
- заказ;
- покупатель;
- вид товара;
- страна-производитель;
- информация о компании.

Инфологическая модель базы данных созданного магазина представлена на рис. 8.

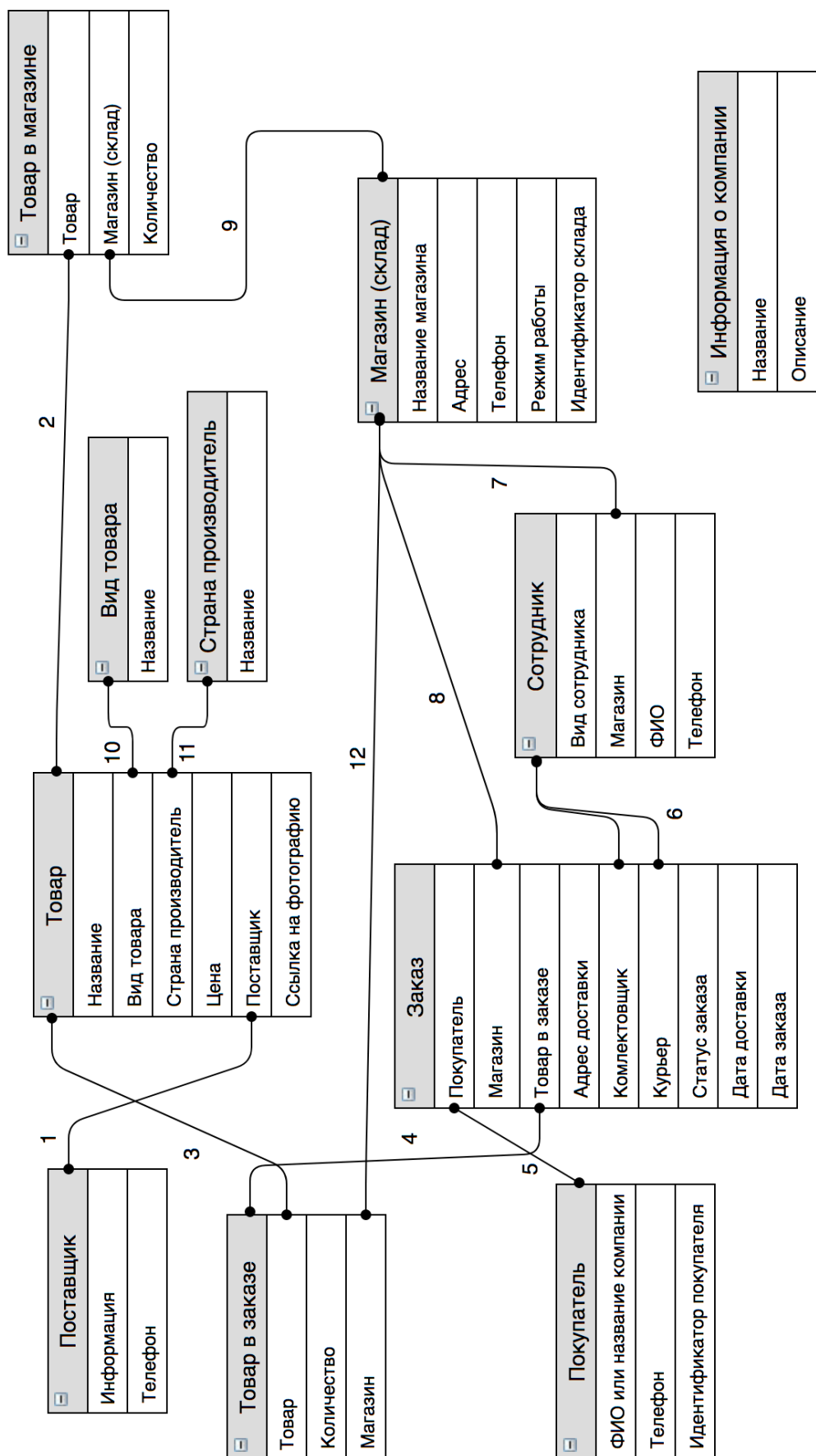


Рис. 8 – Инфологическая модель базы данных

Между сущностями в базе данных было выявлено 11 связей, каждая из которых описана в таблицах 4 и 5 [21, 22].

Номер связи	Сущность 1	Сущность 2	Вид связи
1	Поставщик	Товар	Один ко многим
2	Товар	Товар в магазине	Один ко многим
3	Товар в заказе	Товар	Многие к одному
4	Товар в заказе	Заказ	Многие ко многим
5	Заказ	Покупатель	Многие к одному
6	Сотрудник	Заказ	Один ко многим
7	Магазин (склад)	Сотрудник	Один ко многим
8	Заказ	Магазин (склад)	Многие к одному
9	Магазин (склад)	Товар в магазине	Один ко многим
10	Товар	Вид товара	Многие к одному
11	Товар	Страна-производитель	Многие к одному

Таблица 4 – Виды связей между сущностями

Номер связи	Сущность 1	Сущность 2	Модальность связи
1	Поставщик	Товар	Может - Должен
2	Товар	Товар в магазине	Может - Должен
3	Товар в заказе	Товар	Должен - Может
4	Товар в заказе	Заказ	Должен - Может
5	Заказ	Покупатель	Должен - Должен
6	Сотрудник	Заказ	Может - Должен
7	Магазин (склад)	Сотрудник	Должен - Должен
8	Заказ	Магазин (склад)	Может - Может
9	Магазин (склад)	Товар в магазине	Может - Должен
10	Товар	Вид товара	Должен - Может
11	Товар	Страна-производитель	Должен - Может

Таблица 5 – Модальности связей между сущностями

Для выделенных сущностей и связей, с учетом их видов и модальностей, а также проведя нормализацию, можно построить схему базы данных. Схема базы данных, с указанием типов данных атрибутов сущностей представлена на рис. 9 [20, 21].

В таблице 6 представлено соответствие наименований сущностей инфологической модели базы данных и имен полей схемы базы данных.

Сущности инфологической модели БД	Имена полей схемы БД
Поставщик	providers
Товар	products
Товар в заказе	products in orders
Товар в магазине	products in shops
Магазин (склад)	shops
Сотрудник	clients
Заказ	orders
Покупатель	clients
Вид товара	product_types
Страна-производитель	countries
Информация о компании	about

Таблица 6 – Соответствие сущности инфологической модели БД и имен полей базы данных

Описание каждого атрибута каждой сущности базы данных представлены в таблицах 7 – 17.

Название атрибута	Описание атрибута
title	Название компании-поставщика и/или ИНН
phone	Телефон для связи с представителем поставщика

Таблица 7 – Таблица providers сущности Поставщик

Информация о поставщике вынесена в отдельную сущность, в связи с тем, что у многих товаров может быть один и тот же поставщик и эта информация будет дублироваться, тем самым нарушая нормализацию.

Название атрибута	Описание атрибута
title	Название товара
product_type	Внешний ключ на таблицу product_types, один из возможных типов товара
country	Внешний ключ на таблицу countries, одна из возможных стран-производителей товара
price	Цена товара
provider	Внешний ключ на таблицу providers, один из возможных поставщиков товара
image	Ссылка на изображение товара

Таблица 8 – Таблица products сущности Товар

Название атрибута	Описание атрибута
title	Название одного из видов товара

Таблица 9 – Таблица product_types сущности Вид товара

Название атрибута	Описание атрибута
title	Наименование страны-производителя

Таблица 10 – Таблица countries сущности Страна-производитель

Информация о виде товара и стране производителе вынесена в отдельные сущности по нескольким причинам. Во-первых, это позволяет оставаться базе данных во 2 нормальной форме. Во-вторых, по этим параметрам происходит фильтрация товара на сайте-каталоге и в административной части.

Название атрибута	Описание атрибута
product	Внешний ключ на таблицу products, товар
shop	Внешний ключ на таблицу shop, магазин в котором находится товар
count	Количество данного товара в этом магазине

Таблица 11 – Таблица products_in_shops сущности Товар в магазине

Название атрибута	Описание атрибута
product	Внешний ключ на таблицу products, товар
count	Количество данного товара в заказе

Таблица 12 – Таблица products_in_orders сущности Товар в заказе

Название атрибута	Описание атрибута
data	ФИО клиента или наименование юридического лица
phone	Телефон клиента
legal_entity	Указатель на тип клиента: юридическое или физическое лицо. По умолчанию – физическое.

Таблица 13 – Таблица clients сущности Покупатель

Название атрибута	Описание атрибута
title	Название магазина или склада
address	Адрес магазина
phone	Телефон магазина
worktime	Режим работы магазина
stock	Идентификатор склада

Таблица 14 – Таблица shops сущности Магазин

Название атрибута	Описание атрибута
employee_type	Тип сотрудника (занимаемая должность в компании)
shop	Внешний ключ на таблицу shops, магазин (или склад) к которому привязан данный сотрудник
user	ФИО сотрудника
phone	Телефон сотрудника

Таблица 15 – Таблица employees сущности Сотрудник

Название атрибута	Описание атрибута
title	Название компании
text	Описание компании

Таблица 16 – Таблица about сущности Информация о компании

Название атрибута	Описание атрибута
client	Внешний ключ на таблицу clients, клиент, который оформил заказ
shop	Внешний ключ на таблицу shops, к которому привязан данный заказ
product_in_order	Внешний ключ на таблицу product_in_orders, товары, которые находятся в заказе
delivery_address	Адрес доставки
supplier	Внешний ключ на таблицу employees, комплектовщик заказа
courier	Внешний ключ на таблицу employees, водитель-курьер, доставляющий заказ
status	Статус заказа
create_date	Дата создания заказа
delivery_date	Дата передачи заказа клиенту

Таблица 17 – Таблица orders сущности Заказ

Спроектированная база данных находится в 3 нормальной форме, так как: значения всех атрибутов всех сущностей атомарны; все атрибуты, не входящие в первичный ключ, связаны с ним полной функциональной зависимостью; ни один неключевой атрибут функционально не зависит от любого другого неключевого атрибута [22].

2.4. Описание функционала пользователей

Работать с созданным при помощи приложения интернет-магазином смогут пользователи двух видов:

- внутренний пользователь (работник компании);
- внешний пользователь (покупатель).

Описание функционала внутреннего пользователя

В административной части созданного интернет-магазина выделены 5 типов внутренних пользователей:

- администратор компании;
- менеджер компании;
- менеджер определенного магазина или склада;
- работник определенного магазина или склада;
- водитель-курьер.

Описание функционала и ограничений для каждого типа внутреннего пользователя представлено ниже.

Водителю-курьеру доступны заказы со статусами «Ожидает доставки», «В пути».

Водитель-курьер может:

- просматривать заказы со статусом «Ожидает доставки» и необходимую информацию о заказе: информацию о клиенте, телефон клиента, товары в заказе, адрес доставки, а также ФИО комплектовщика заказа;
- менять статус заказа с «Ожидает доставки» на «В пути», в том случае, если водитель-курьер решил взять этот заказ на доставку;
- менять статус заказа на «Завершен», в том случае, если он доставил заказ клиенту.

Работнику определенного магазина доступны заказы своей торговой точки со статусами «Заказ комплектуется» и «Собран, готов к выдаче». Помимо этого, сотрудник имеет информацию о количестве товара в своей торговой точке.

Работнику склада, в свою очередь, доступны заказы со статусом «Заказ комплектуется» и «Ожидает доставки». Кроме этого, есть информация о количестве товара на складе.

Работники магазина и склада могут:

- просматривать заказы со статусом «Заказ комплектуется» и «Собран, готов к выдаче» («Ожидает доставки» для работников склада), а также необходимую информацию о заказе: информацию о клиенте, телефон клиента, товары в заказе, адрес доставки;
- менять статус заказа на «Собран, готов к выдаче» («Собран, ожидает доставки» для работников склада) и «Завершен» (только для работников магазина, так как на складе заказ со статусом «Ожидает доставки» в дальнейшем будет обрабатывать водитель-курьер);
- добавлять товар и изменять его количество в магазине или на складе.

Менеджеру магазина (склада) доступны заказы с любым статусом своего магазина (склада), а также имеется доступ к информации о товарах и сотрудниках магазина (склада).

Менеджеры магазина (склада) могут:

- использовать все функции работника магазина (склада);
- просматривать заказы своего магазина (склада) любым статусом, а также получать информацию о комплектовщике заказа и курьере, который доставил этот заказ;
- просматривать информацию о сотрудниках своего магазина (склада).

Менеджеру компании предоставляются заказы с любым статусом всех магазинов и склада, а также доступна информация о товарах и сотрудниках.

Менеджеры магазина (склада) могут:

- просматривать все заказы с любыми статусами всех магазинов и склада, а также фильтровать их по статусу и магазину (складу);
- просматривать информацию о товарах во всех магазинах и на складе, а также изменять их количество и добавлять новые позиции товара в определенный магазин или на склад;
- просматривать информацию о всех сотрудниках компании.

Администратору предоставляется доступ к полной информации о магазине.

Администратор может:

- создавать интернет-магазин для компании;
- изменять название интернет-магазина и его описание;
- настраивать способ адресной доставки;
- указывать меру товаров и единицы измерения;
- работать со всеми пользователями системы, то есть добавлять новых сотрудников, а также редактировать сведения уже существующих: менять их личную информацию, занимаемую в компании должность и место работы;
- изменять информацию о внешних пользователях системы;
- добавлять новые магазины в систему, указывать их название, адрес и режим работы, а также изменять и удалять существующие;
- изменять и добавлять параметры товаров и их поставщиков.

Описание функционала внешнего пользователя

Всем внешним пользователям предоставляются следующие возможности:

- просмотр общей информации о компании;
- просмотр информации о торговых точках компании и складе:
 - a. название магазина;
 - b. адрес магазина;
 - c. режим работы;
 - d. телефон.
- просмотр электронного каталога товаров компании:
 - a. информация о конкретном товаре:
 - i. название;
 - ii. вид;
 - iii. цена;
 - iv. максимально возможное количество для заказа;
 - v. страна-производитель;
 - vi. фотография товара.
 - b. фильтрация товаров при просмотре каталога товаров:
 - i. по виду;
 - ii. по цене;
 - iii. по стране-производителю.
 - c. сортировка товаров при просмотре каталога товаров:
 - i. по цене;
 - ii. по названию.
- регистрация пользователя в системе с указанием типа пользователя, данных о пользователе, телефона, адреса электронной почты и пароля для входа в личный кабинет. Личный кабинет создается автоматически после регистрации;

- оформление заказа, то есть указание необходимых товаров с их количеством;
- подтверждение заказа, то есть ввод адреса доставки и подтверждение корректности выбора товаров в заказе;
- возможность изменение данных о пользователе в личном кабинете.

В разрабатываемой модели интернет-магазина предусмотрено два вида внешних пользователей: физическое и юридическое лицо. По сути, эти пользователи-клиенты никак не отличаются, но их необходимо отличать. Существуют такие товары, продавать которые через Интернет на территории РФ запрещено, или разрешено только юридическим лицам. Примером такого товара можно считать алкоголь. В рамках интернет-магазина алкоголь физическое лицо сможет оформить только самовывоз товара из определенного магазина, а юридическое лицо может оформить доставку по необходимому адресу прямо со склада.

Глава 3. Особенности реализации приложения с помощью средств Django

3.1. Описание основных алгоритмов реализации программного продукта

Описание структуры директорий и файлов проекта находится в приложении. Ниже следует описание структуры и работы основных файлов разработанного приложения.

Файл *alcohol_shop/urls.py* содержит список url-маршрутов, которые должны обрабатываться. Каждому маршруту соответствует свой класс представления, который и обрабатывает запросы от клиентов, и возвращает данные.

Данные классы представлений хранятся в файле *alcohol_shop/app_alcohol_shop/views.py*. Стоит заметить, что для взаимодействия с клиентами можно использовать функции вместо классов, но в данном случае использование классов является более подходящим решением, так как дает возможность использовать наследование, примеси и другие возможности ООП.

Например, в шапке сайта необходимо отображать данные о интернет-магазине. Если бы использовались функции вместо классов, то для каждой функции необходимо было бы извлекать данные о магазине и передавать в шаблон, что чревато ошибками, да и в дальнейшем если логика изменится, то придется изменять ее во всех функциях. Используя классы, был создан один базовый класс *BaseContentShopMixin*, который хранит в себе данные о магазине и добавляется к другим классам, где это необходимо.

Пример класса отображения главной страницы:

```
class IndexPage(BaseContentShopMixin, BaseClientContentMixin,
TemplateView):
    template_name = 'client/base.html'
```

Как видно из примера, можно использовать множественное наследование, и наследоваться от нескольких классов. В данном случае IndexPage должен вернуть HTML-документ, а за это отвечает класс TemplateView, поэтому в параметре template_name указывается шаблон, который нужно вернуть.

В Django уже реализовано большое количество классов, которые облегчают работу с данными клиентов и базой данных. Список Django-классов, которые используются для создания собственных представлений: View, TemplateView, CreateView, FormView, ListView, UpdateView, DeleteView.

Сущности в базе данных описываются специальными классами, называемыми моделями (models.Model), которые хранятся в директории *alcohol_shop/app_alcohol_shop/models.py*. В классах представлений Django уже реализованы методы работы с базой данных, поэтому необходимо лишь указывать название класса модели, с которой данный класс должен работать. Например, необходимо отобразить список товаров, тогда для этого создается следующий класс:

```
class ProductListView(BaseContentShopMixin, ListView):
    model = models.ProductInShop
    paginate_by = 9
    template_name = 'client/product_list.html'
```

В данном классе снова использовался общий класс BaseContentShopMixin для отображения информации о магазине и Django-класс ListView. В ListView уже реализован метод для отображения списка объектов из базы данных. Все, что необходимо сделать – передать название класса модели, данные которой будут извлекаться, (model = models.ProductInShop) и название шаблона (template_name = 'client/product_list.html'). Также, если есть необходимость постраничного вывода информации, то указывается параметр (paginate_by = 9), который определяет какое количество объектов будет отображаться на одной странице.

Пример класса, проверяющий авторизацию клиента, который будет подмешиваться в те классы, в которых необходимо, чтобы клиент был авторизован:

```
class ClientLoginMixin(View):  
    @method_decorator(login_required(login_url='/'))  
    def dispatch(self, request, *args, **kwargs):  
        try:  
            request.user.client_profile  
        except:  
            return HttpResponseRedirect()  
        return super().dispatch(request, *args, **kwargs)
```

В данном классе оборачивается метод dispatch декоратором login_required. Если клиент не авторизован, он перенаправляется на главную страницу, то есть по url-маршруту стартовой страницы. Также добавляется проверка, является ли авторизованный пользователь клиентом.

Класс для обновления личной информации клиентом наследуется от класса UpdateView, который используется при обновлении данных:

```
class PersonalArea(BaseClientContentMixin, UpdateView):  
    model = models.Client  
    success_url = reverse_lazy('index_page')  
    form_class = forms.PersonalAreaForm  
    template_name = 'client/personal_area.html'
```

`success_url = reverse_lazy('index_page')` – указывается тот url-маршрут, на который будет перенаправлен клиент при успешном выполнении;

`form_class = forms.PersonalAreaForm` – указывается класс формы персональной информации.

Необходимо переопределить метод `get`, чтобы личная информация отображалась тому клиенту, которому она принадлежит:

```
def get(self, request, *args, **kwargs):  
    if int(kwargs.get('pk')) != int(self.request.user.id):  
        return redirect(reverse_lazy('index_page'))  
    kwargs['pk'] = self.request.user.client_profile.id  
    self.kwargs['pk'] = self.request.user.client_profile.id  
    return super().get(request, *args, **kwargs)
```

Также необходимо переопределить метод `post`, чтобы личная информация обновлялась для того клиента, которому она принадлежит.

```
def post(self, request, *args, **kwargs):  
    if int(kwargs.get('pk')) != int(self.request.user.id):  
        return redirect(reverse_lazy('index_page'))  
    kwargs['pk'] = self.request.user.client_profile.id  
    self.kwargs['pk'] = self.request.user.client_profile.id  
    return super().post(request, *args, **kwargs)
```

В приложении создаются собственные классы фильтрации. Класс фильтрации товаров `ProductFilter` наследуется от `django_filters.FilterSet`:

```

class ProductFilter(django_filters.FilterSet):

    price_min = django_filters.NumberFilter(name='product__price',
lookup_expr='gt')
    price_max = django_filters.NumberFilter(name='product__price',
lookup_expr='lt')

    class Meta:
        model = models.ProductInShop
        fields = ['product__price', 'product__title', 'product__product_view',
'product__product_country', 'shop']

```

В `model` указывается модель, для которой будет осуществляться фильтрация, а в `Fields` список полей, по которым она будет осуществляться. Если необходимо указать для фильтрации поле из другой модели, с которой есть связь, то это делается через двойное подчеркивание.

Например, `product__price` – это поле `price` из модели, с которой связано поле `product`.

`price_min = django_filters.NumberFilter(name='product__price', lookup_expr='gt')` – собственное поле фильтрации минимальной цены, где `lookup_expr='gt'` означает «больше чем», а `lookup_expr='lt'` – «меньше чем». Для того, чтобы использовать классы фильтрации в представлении необходимо добавить параметр `filterset_class = ProductFilter`.

Класс для создания заказов `AddProductToOrder(CreateView)` наследуется от `CreateView`, переопределяется метод `post`. Проверяется, есть ли неоформленный заказ. Если есть, то класс получает его, если нет, то создается новый.

```

if models.Order.objects.filter(client=self.request.user.client_profile,
status__in=['ACTIVE', 'NOPRODUCTS']).count() == 0:

```

```
order = models.Order.objects.create(client=self.request.user.client_profile,  
shop=product_in_shop.shop)  
  
else:  
order = models.Order.objects.filter(client=self.request.user.client_profile,  
status__in=['ACTIVE', 'NOPRODUCTS']).first()
```

После этого проверяется достаточно ли товара в магазине, если достаточно, то он добавляется в заказ, и количество товара в магазине уменьшается на заказанное количество.

```
if product_in_shop.count >= count:  
    product_in_order =  
models.ProductInOrder.objects.create(product=product_in_shop.product,  
count=count)  
    order.product_in_order.add(product_in_order)  
    order.save()  
    product_in_shop.count = product_in_shop.count - count  
    product_in_shop.save()
```

Класс удаления заказов наследуется от класса DeleteView, в котором реализован механизм удаления данных и от класса EmployeeLoginMixin, в котором реализована авторизация сотрудника.

```
class OrderDelete(EmployeeLoginMixin, DeleteView):  
  
    model = models.Order  
    success_url = reverse_lazy('employee_orders')
```



```

template_name = 'employee/order_confirm_delete.html'

def delete(self, request, *args, **kwargs):
    if self.request.user.employee_profile.employee_type not in
[models.Employee.ROLE_ADMIN,
models.Employee.ROLE_MANAGER_COMPANY,

models.Employee.ROLE_MANAGER_SHOP]:
        return redirect(reverse_lazy('employee_orders'))
    return super().delete(request, *args, **kwargs)

```

Также переопределяется метод `delete`, в котором реализована проверка на наличие необходимых прав у сотрудника. Если у сотрудника таких прав нет, то его перенаправляет к списку заказов.

Далее рассматривается шаблон отображения списка товаров, находящийся по пути: *templates/client/product_list.html*. Во всех шаблонах используется механизм наследования.

`{% extends 'client/base.html' %}` указывает на то, что данный шаблон наследуется от базового, и в нем будет отображена верстка из базового шаблона.

Так же в шаблонах используется модуль для работы с `bootstrap4`, который необходимо подключить следующей командой:

```
{% load bootstrap4 %}
```

Если есть необходимость отображать пагинацию, то выводится следующий блок:

```

{% if is_paginated %}
<nav aria-label="Page navigation example">
  <ul class="pagination">
    {% if page_obj.has_previous %}

```

```

        <li class="page-item"><a class="page-link" href="?page={{
page_obj.previous_page_number }} {% for key, value in request.GET.items %} {%
if key != 'page' %}&{{ key }}={{value}} {% endif %} {% endfor
%}">&laquo;</a></li>

        {% endif %}

        {% for page in paginator.page_range %}

            <li class="page-item" {% if page_obj.number == forloop.counter
%}>active{% endif %}"><a class="page-link" href="?page={{forloop.counter}} {%
for key, value in request.GET.items %} {% if key != 'page' %}&{{ key
}}={{value}} {% endif %} {% endfor %}">{{forloop.counter}}</a></li>

            {% endfor %}

            {% if page_obj.has_next %}

                <li class="page-item"><a class="page-link" href="?page={{
page_obj.next_page_number }} {% for key, value in request.GET.items %} {% if
key != 'page' %}&{{ key }}={{value}} {% endif %} {% endfor
%}">&raquo;</a></li>

            {% endif %}

        </ul>

    </nav>

{% endif %}

```

{% for key, value in request.GET.items %} — добавление в строку параметров запросов, это необходимо для того, чтобы при переходе на вторую страницу списка фильтрованных товаров не исчезала выбранная ранее фильтрация товаров.

3.2. Описание разработанных статусов заказа

Всего в разработанной системе предусмотрены следующие статусы для заказов:

- «Заказ комплектуется»;
- «Собран, готов к выдаче»;
- «Ожидает доставки»;
- «В пути»;
- «Завершен».

Статус «Заказ комплектуется» заказ получает сразу же после того, как пользователь-покупатель подтверждает корректность собранного заказа в личном кабинете. После этого заказ появляется в системе заказов для просмотра и сборки необходимыми сотрудниками компании. Далее, если заказ укомплектован, то сотрудник меняет его статус на «Собран, готов к выдаче» или «Ожидает доставки». Статус «Собран, готов к выдаче» ставится в том случае, если заказ оформлен самовывозом из определенного магазина и ожидает клиента. Статус «Ожидает доставки» ставится, если необходимо доставить заказ по необходимому адресу и ожидает курьера-водителя, который осуществляет доставку. После того как водитель-курьер забрал заказ со склада для его доставки, он меняет статус заказа на «В пути». Данный статус позволяет сотрудникам и клиенту контролировать заказ. Например, если клиент видит, что заказ имеет статус «В пути», то следует ожидать доставки в ближайшее время. Статус «Завершен» присваивается заказу, когда заказ был передан или доставлен клиенту.

3.3. Описание процесса создания интернет-магазина

Процесс управления и поддержки интернет-магазина начинается с его создания. Для этого пользователю, который занимает должность администратора интернет-магазина в компании, необходимо в браузере на любом устройстве перейти по ссылке <http://www.daleonn.store>. В окне браузера пользователь увидит форму создания магазина (рис. 10). Для создания необходимо заполнить поля: электронная почта администратора, пароль и доменное имя. Связка электронной почты и пароля необходима пользователю для дальнейшего входа в систему, поэтому ее необходимо запомнить. Для этого эта связка будет выведена на экран еще раз, сразу после создания магазина (рис. 11).

Все магазины, созданные с помощью веб-приложения, будут располагаться на доменах второго уровня, то есть на поддоменах домена daleonn.store. Таким образом, пользователь может выбрать только начало адреса магазина. Адрес любого созданного в данной системе магазина будет выглядеть следующим образом: *.daleonn.store, где * – введенное при создании интернет-магазина доменное имя.

У пользователя есть возможность посмотреть уже созданные другими компаниями интернет-магазины, их адреса расположены в правой части окна. Это необходимо для того, чтобы пользователь понимал, как будет выглядеть его будущий магазин. После введенных данных, необходимых при регистрации, и нажатия кнопки «Добавить» появится окно с введенными данными, созданный магазин добавится в «Список магазинов» (рис. 11). Теперь пользователь может переходить по указанному доменному имени и приступать к настройке и комплектации интернет-магазина. Все эти процессы будут описаны в параграфе 3.4.

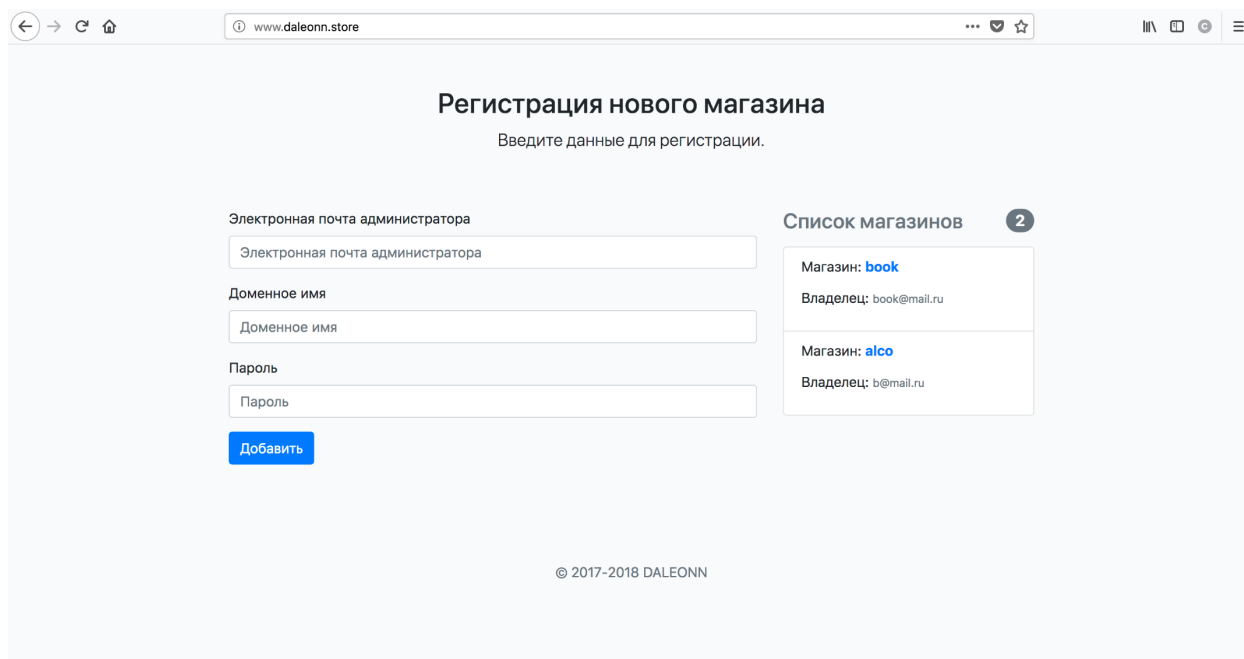


Рис. 10 – Интерфейс главной страницы приложения. Создание нового интернет-магазина

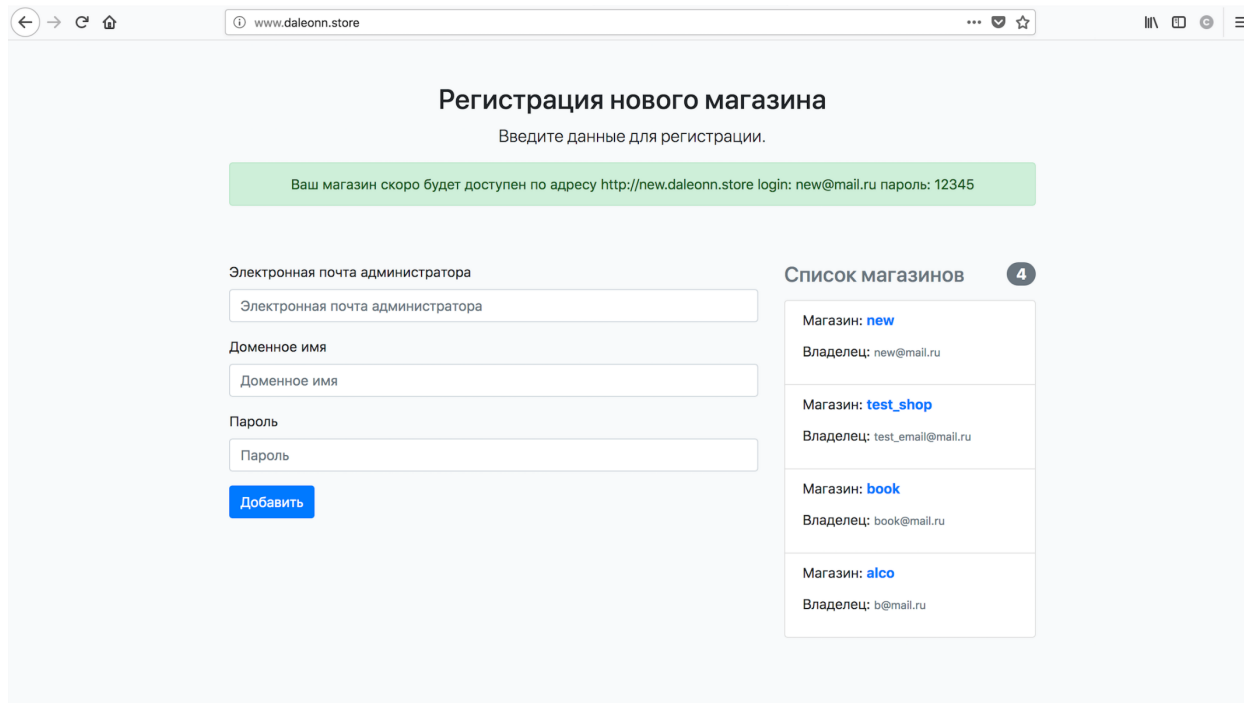


Рис. 11 – Окно создания нового интернет-магазина после ввода необходимых данных

3.4. Описание работы пользователей в созданном интернет-магазине

Процесс создания интернет-магазина описан в параграфе 3.3. Описание всех процессов производится на примере созданного интернет-магазина, занимающегося продажей алкоголя. Этот магазин располагается по адресу <http://alco.daleonn.store>.

Сначала следует описание функционала и работы внешнего пользователя (покупателя) в созданном интернет-магазине компании.

Первым этапом пользователю необходимо попасть на сайт-каталог, перейдя по ссылке интернет-магазина конкретной компании в любом браузере на компьютере или мобильном телефоне. Он увидит экран с отображением всех магазинов компании, а также необходимую информацию о них (рис. 12).

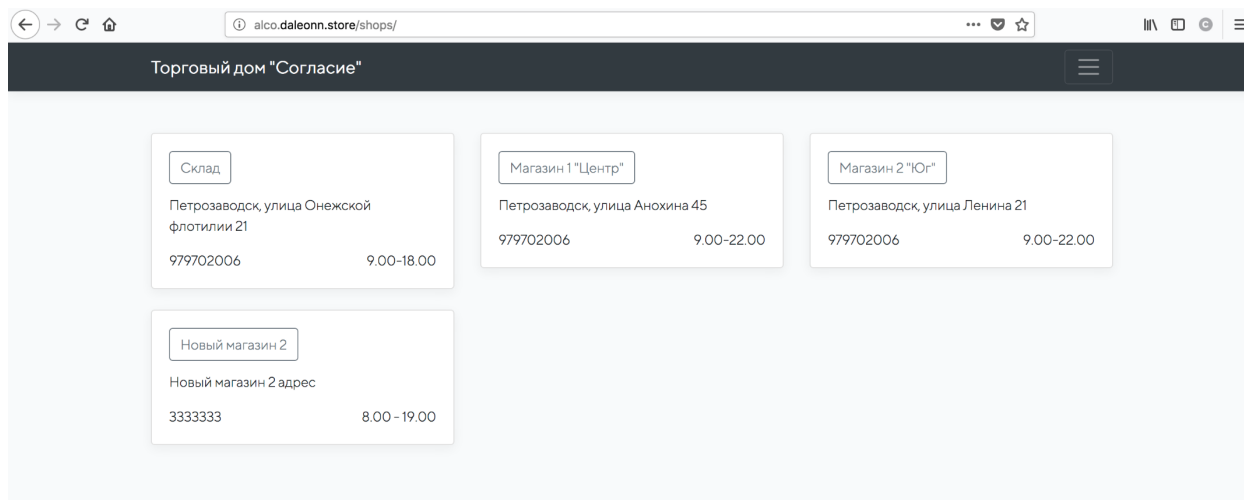


Рис. 12 – Стартовая страница созданного интернет-магазина

После этого, выбрав один из магазинов, он может просматривать ассортимент товаров в данной торговой точке. На страницах магазинов

отображена вся необходимая информация о товарах. Существует возможность осуществлять сортировку товаров по названию и цене (по возрастанию и убыванию) при выводе их на страницу, а также производить фильтрацию товаров по цене, виду товара, стране-производителю и искать товар по названию (рис. 13).

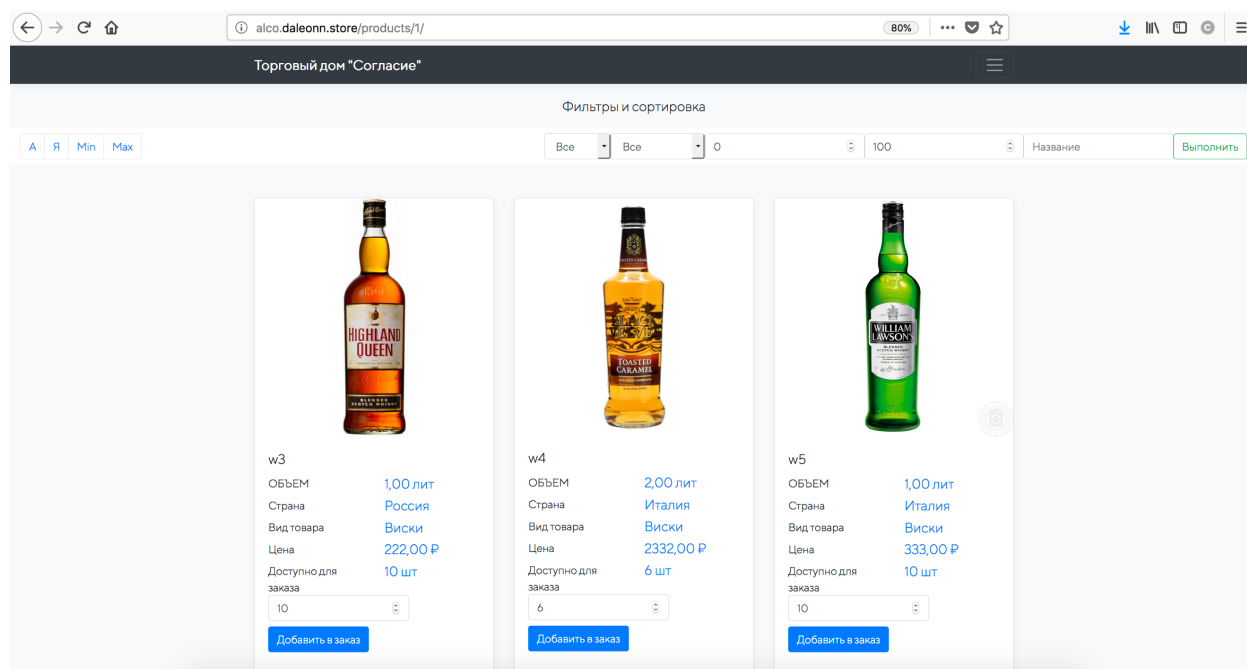


Рис. 13 – Отображение электронного каталога товаров конкретной торговой точки

Чтобы оформить какой-либо заказ, покупателю сначала необходимо зарегистрироваться в системе, указав в процессе регистрации необходимые данные. Форма регистрации показана на рис. 14.

После регистрации клиент входит в кабинет, с указанием электронной почты и пароля, указанных при регистрации, и может осуществлять заказ товаров. Просматривать все заказы и отслеживать их статус клиент может во вкладке «Заказы» (рис. 15).

Менять информацию о себе он может во вкладке «Личный кабинет». После того, как пользователь сформировал заказ, он должен его подтвердить, нажав кнопку «Подтвердить заказ» для определенного заказа.

Регистрация

☐ Юридическое лицо

Ф.И.О или название компании

Ф.И.О или название компании

Телефон

Телефон

Email

Email

Password

Password

Регистрация

Рис. 14– Форма регистрации клиентов компании

Торговый дом "Согласие" Здравствуйте yur@mail.ru

Заказ номер	Дата	Сумма	Вещи	Адрес доставки	Статус
Заказ номер 30	23 апреля 2018 г.	Сумма 424,00 ₽	W1 2 шт	None	Завершен
Заказ номер 37	26 апреля 2018 г.	Сумма 1110,00 ₽	w2 5 шт	None	Заказ комплектуется
Заказ номер 38	26 апреля 2018 г.	Сумма 424,00 ₽	W1 2 шт	Город Петроф	В пути
Заказ номер 39	26 апреля 2018 г.	Сумма 4228,00 ₽	w3 4 шт w6 10 шт	Санкт-Петербург	Ожидает доставки

Рис. 15 – Отображение заказов покупателя в личном кабинете

Далее следует описание работы и функционала каждого типа внутреннего пользователя сервисного приложения.

Водитель-курьер

Водителю-курьеру выдается логин и пароль для входа в систему администратором компании. Используя эту связку, он, как и внешний пользователь, переходит по адресу конкретного магазина и нажимает на кнопку «Вход».

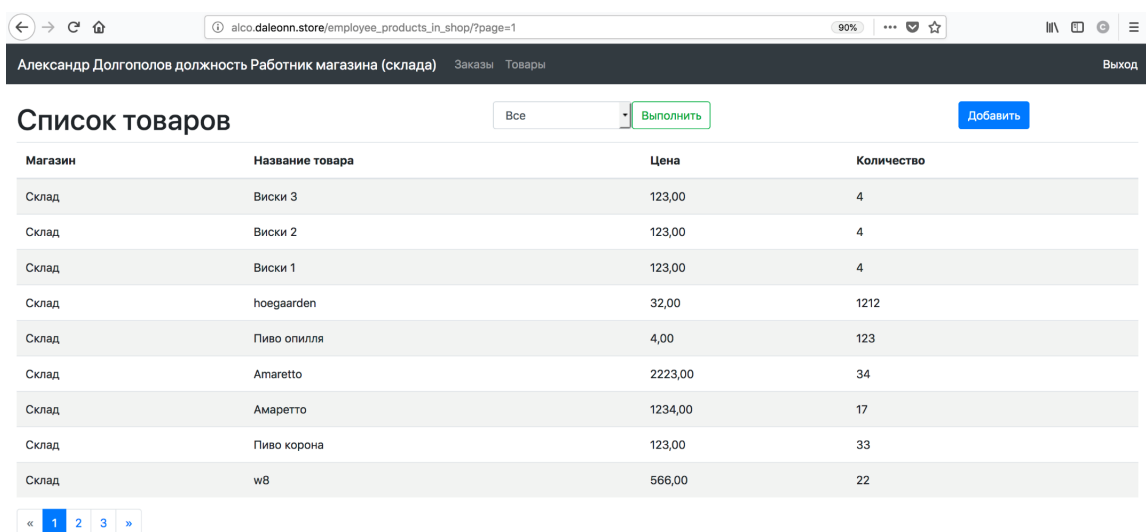
После введенной связки логина и пароля сотрудник автоматически попадает в рабочую область – рис. 16, где он может смотреть необходимые заказы и информацию о них, менять их статус и фильтровать заказы по статусу. Стоит отметить, что отображаются только актуальные заказы для доставки, после того как курьер их доставит, они не будут видны. В результате изменения статуса заказа на «В пути» в графе Курьер отображаются имя и фамилия курьера, который доставил заказ. Это необходимо для того, чтобы администратор, а также другие пользователи системы видели, кто доставлял определенный заказ, чтобы в случае возникновения проблем обращаться именно к этому сотруднику.

№	Клиент	Магазин	Адрес доставки	Товары в заказе	Комплектовщик	Курьер	Статус заказа	Дата заказа	Дата доставки	Действия
#39	yur@mail.ru	Склад	Санкт-Петербург	w3 - 1,00 л - 4 шт 222,00 руб w6 - 2,00 л - 10 шт 334,00 руб Сумма 4228,00 руб		-	Ожидает доставки	26 апреля 2018 г.		Изменить статус
#38	yur@mail.ru	Склад	Город Петроф	W1 - 1,00 л - 2 шт 212,00 руб Сумма 424,00 руб		Константин Малышев	В пути	26 апреля 2018 г.	26 апреля 2018 г.	Изменить статус

Рис. 16– Отображение заказов для Водителя-курьера

Работник магазина (склада)

Работнику магазина или склада выдается логин и пароль для входа в систему администратором компании. Используя эту связку, он переходит по ссылке своего магазина и нажимает кнопку «Вход». После введенной связки логина и пароля сотрудник автоматически попадает в свою рабочую область. Интерфейс рабочей области работника магазина или склада показан на рис. 18. Ему доступны вкладки «Заказы» и «Товары». Во вкладке «Заказы» он может посмотреть всю необходимую информацию о заказах для сборки и менять их статус. После изменения статуса на «Собран, ожидает клиента» имя и фамилия комплектовщика появится в соответствующей графе. Это необходимо для того, чтобы администратор, а также другие пользователи системы видели кто собирал определенный заказ и передавал его клиенту или водителю, чтобы в случае возникновения проблем, например, если в заказ был положен не тот товар или какая-то позиция пропущена, обращаться именно к этому сотруднику. Во вкладке «Товары» (рис. 17) работник видит полный список товаров своей торговой точки, а также может добавлять товар в магазин, нажав кнопку Добавить. Это может произойти в связи с различными ситуациями, например, если клиент захотел вернуть товар.



Магазин	Название товара	Цена	Количество
Склад	Виски 3	123,00	4
Склад	Виски 2	123,00	4
Склад	Виски 1	123,00	4
Склад	hoegaarden	32,00	1212
Склад	Пиво опилля	4,00	123
Склад	Amaretto	2223,00	34
Склад	Амаретто	1234,00	17
Склад	Пиво корона	123,00	33
Склад	w8	566,00	22

Рис. 17 – Отображение вкладки «Товары» для работника магазина (склада)

←

→

↺

🏠

🔍

alco.daleonn.store/employee_orders/

90%

...

📧

📌

🏠

📄

🔄

⋮

Выход

Александр Долгополов должность Работник магазина (склада)

Заказы

Товары

Список заказов

Все

Выполнить

№	Клиент	Магазин	Адрес доставки	Товары в заказе	Комплектовщик	Курьер	Статус заказа	Дата заказа	Дата доставки	Действия
#39	yur@mail.ru	Склад	Санкт-Петербург	<div>w3 - 1,00 л - 4 шт 222,00 руб</div> <div>w6 - 2,00 л - 10 шт 334,00 руб</div> <div>Сумма 4228,00 руб</div>	Ольга Завьялова	-	Ожидает доставки	26 апреля 2018 г.		Изменить статус
#37	yur@mail.ru	Склад	None	<div>w2 - 1,00 л - 5 шт 222,00 руб</div> <div>Сумма 1110,00 руб</div>		-	Заказ комплектуется	26 апреля 2018 г.		Изменить статус
#27	tx-tx@mail.ru	Склад	Самовывоз	<div>w3 - 1,00 л - 6 шт 222,00 руб</div> <div>Виски 7 - 1,00 л - 1 шт 123,00 руб</div> <div>Сумма 1455,00 руб</div>	Ольга Завьялова	Константин Малышев	Заказ комплектуется	19 апреля 2018 г.	22 апреля 2018 г.	Изменить статус
#23	tx-tx@mail.ru	Склад	Самовывоз	<div>23r32 - 32,00 л - 2 шт 322,00 руб</div> <div>Сумма 644,00 руб</div>		-	Заказ комплектуется	15 апреля 2018 г.		Изменить статус
#22	tx-tx@mail.ru	Склад	Самовывоз	<div>Виноя - 1,00 л - 1 шт 222,00 руб</div> <div>Сумма 222,00 руб</div>		-	Заказ комплектуется	15 апреля 2018 г.		Изменить статус

Рис. 18 – Отображение личного кабинета работника магазина (склада)

Менеджер магазина (склада)

Менеджеру магазина или склада выдается логин и пароль для входа в систему администратором компании. Используя эту связку, он переходит по ссылке своего магазина и нажимает кнопку «Вход».

После введенной связки логина и пароля сотрудник автоматически попадает в свою рабочую область. Интерфейс рабочей области менеджера магазина показан на рис. 19. Помимо вкладок «Заказы» и «Товары» ему также доступна вкладка «Сотрудники», где он может посмотреть список всех сотрудников своего магазина или склада – рис. 20.

Во вкладке «Заказы» (рис. 19) менеджеру доступны все заказы своего магазина, в том числе со статусом «Завершен», а также есть возможность удаления заказа и его изменения (рис. 20). Вкладка «Товары» идентична вкладке «Товары» для сотрудника компании, имеющего должность работник магазина или склада.

<div> <div> <div>←</div> <div>→</div> <div>🏠</div> </div> <div>alco.daleonn.store/employee_orders/</div> <div> <div>90%</div> <div>...</div> <div>📧</div> <div>🌟</div> </div> <div> <div>🖨</div> <div>📄</div> <div>⌂</div> <div>☰</div> </div> </div> <div> <div>Андрей Бенагуев должность Менеджер магазина (склада)</div> <div>Заказы</div> <div>Товары</div> <div>Сотрудники</div> <div>Выход</div> </div>										
<div> <div>Список заказов</div> <div> <div>Все</div> <div>Выполнить</div> </div> </div>										
№	Клиент	Магазин	Адрес доставки	Товары в заказе	Комплектовщик	Курьер	Статус заказа	Дата заказа	Дата доставки	Действия
#39	yur@mail.ru	Склад	Санкт-Петербург	w3 - 1,00 л - 4 шт 222,00 руб	Ольга Завьялова	-	Ожидает доставки	26 апреля 2018 г.		<div>Изменить заказ</div> <div>Удалить</div>
				w6 - 2,00 л - 10 шт 334,00 руб						
				Сумма 4228,00 руб						
#38	yur@mail.ru	Склад	Город Петроф	W1 - 1,00 л - 2 шт 212,00 руб	Ольга Завьялова	Константин Малышев	В пути	26 апреля 2018 г.	26 апреля 2018 г.	<div>Изменить заказ</div> <div>Удалить</div>
				Сумма 424,00 руб						
#37	yur@mail.ru	Склад	None	w2 - 1,00 л - 5 шт 222,00 руб		-	Заказ комплектуется	26 апреля 2018 г.		<div>Изменить заказ</div> <div>Удалить</div>
				Сумма 1110,00 руб						
#30	yur@mail.ru	Склад	None	W1 - 1,00 л - 2 шт 212,00 руб	Ольга Завьялова	Константин Малышев	Завершен	23 апреля 2018 г.	23 апреля 2018 г.	<div>Изменить заказ</div> <div>Удалить</div>
				Сумма 424,00 руб						
#27	tx-tx@mail.ru	Склад	Самовывоз	w3 - 1,00 л - 6 шт 222,00 руб	Ольга Завьялова	Константин Малышев	Заказ комплектуется	19 апреля 2018 г.	22 апреля 2018 г.	<div>Изменить заказ</div> <div>Удалить</div>
				Виски 7 - 1,00 л - 1 шт 123,00 руб						
				Сумма 1455,00 руб						

Рис. 19 – Интерфейс рабочей области для менеджера магазина или склада

<div> <div> <div>←</div> <div>→</div> <div>🏠</div> </div> <div>alco.daleonn.store/order_status_update/39/</div> <div> <div>90%</div> <div>...</div> <div>📧</div> <div>🌟</div> </div> <div> <div>🖨</div> <div>📄</div> <div>⌂</div> <div>☰</div> </div> </div> <div> <div>Андрей Бенагуев должность Менеджер магазина (склада)</div> <div>Заказы</div> <div>Товары</div> <div>Сотрудники</div> <div>Выход</div> </div>										
<div> <div>Статус заказа</div> <div>Ожидает доставки</div> </div>										
<div> <div>Адрес доставки</div> <div>Санкт-Петербург</div> </div>										
<div> <div>Удалить товары из заказа</div> <div> <div>w3 Поставщик 1 979702006 Виски Россия 1.00 222.00 images/w3.jpg 4</div> <div>w6 Поставщик 2 979702006 Виски Россия 2.00 334.00 images/w6.jpg 10</div> </div> </div>										
<div> <div>Добавить товары в заказ</div> <div> <div>-----</div> </div> </div>										
<div> <div>Количество товаров</div> <div>Количество товаров</div> </div>										
<div> <div>Изменить</div> </div>										

Рис. 20 – Интерфейс изменения заказа

Имя	Магазин	Телефон	Должность
Кирилл Иванов	Склад	979702006	Водитель-курьер
Иван Тамбовцев	Склад	89213233435	Работник магазина (склада)
Елена Петрова	Склад	894385934	Менеджер магазина (склада)
Константин Малышев	Склад	89213233434	Водитель-курьер
Никита Филимонов	Склад	+79992223344	Работник магазина (склада)
Александр Долгополов	Склад	8932938923	Работник магазина (склада)
Андрей Бенагуев	Склад	83948398	Менеджер магазина (склада)

Рис. 21 – Отображение списка сотрудников

Менеджер компании

Менеджеру компании выдается логин и пароль для входа в систему администратором компании. Используя эту связку, он переходит по ссылке интернет-магазина и нажимает кнопку «Вход». После этого сотрудник автоматически попадает в свою рабочую область. Интерфейс рабочей области менеджера компании показан на рис. 22.

Менеджеру компании доступны вкладки «Заказы», «Товары» (рис. 23), «Сотрудники» (рис. 27) и «Клиенты» (рис. 24), в последней из которых он может смотреть всю информацию о клиентах компании, то есть о зарегистрированных в данном магазине внешних пользователях. Отличие от всех предыдущих пользователей заключается в том, что менеджер магазина может смотреть заказы, товары и сотрудников сразу всех магазинов, а также вносить туда изменения. Помимо этого, у менеджера компании есть возможность фильтровать товары и сотрудников по магазинам, в которых они работают.

Менеджер компании имеет возможность выгрузки каталога товаров в систему путем добавления файла, в котором указана вся необходимая информация о товарах и их поставщиках. Это необходимо для того, чтобы

уменьшить время выгрузки большого списка товаров в систему, так как выгрузка товаров поштучно занимает достаточно много времени.

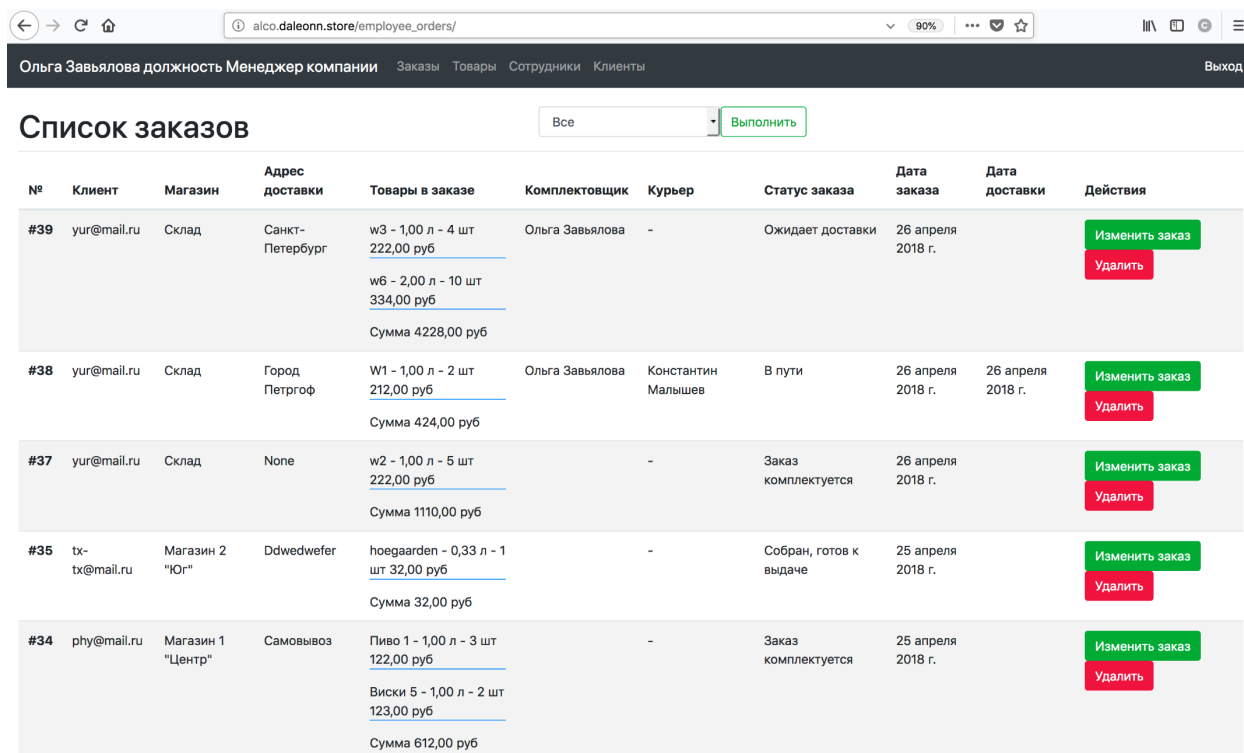


Рис. 22 – Интерфейс рабочей области для менеджера компании

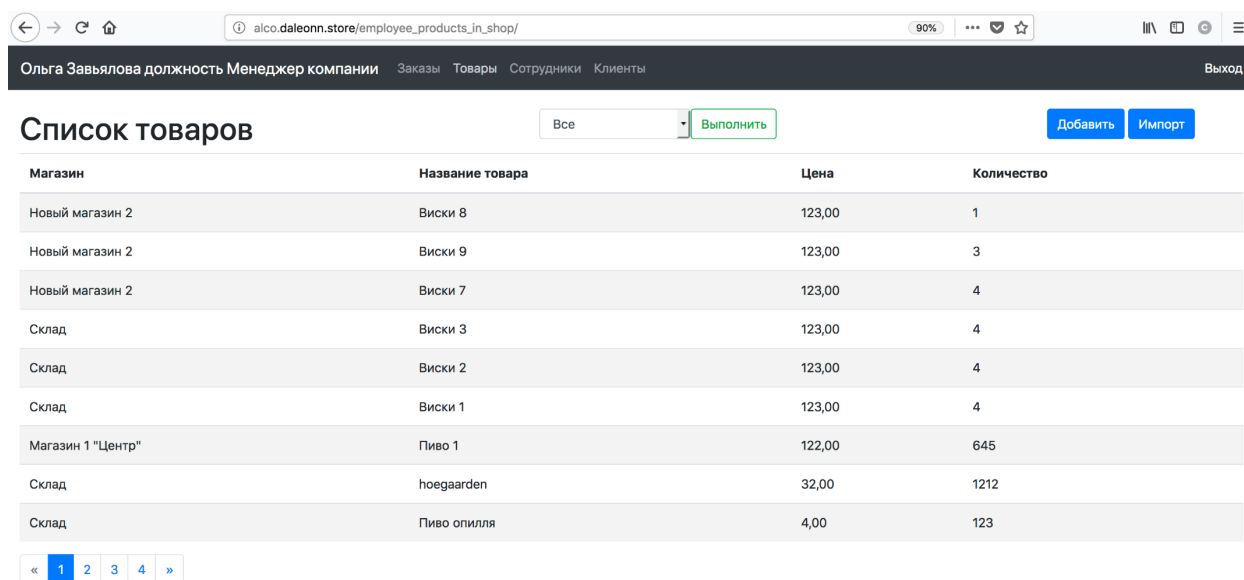


Рис. 23 – Отображение вкладки «Товары» для менеджера компании

Email	Данные	Телефон	Тип клиента
tx-tx@mail.ru	vevewve	979702006	Частное лицо
tx-tx2@mail.ru	werfwefewfrell88	979702006	Юридическое лицо
yur@mail.ru	fjiff	498593583	Юридическое лицо
phy@mail.ru	edwdedwQQ	+7 925 888 83 13	Частное лицо

« 1 »

Рис. 24– Отображение вкладки «Клиенты» для менеджера компании

Администратор компании

Администратор компании для входа в свою рабочую область использует связку электронной почты и логина, введенных при создании интернет-магазина. Сотрудник переходит по ссылке, указанной при создании магазина, и вводит эту связку, после чего автоматически попадает в свою рабочую область. Сразу после создания магазина эта рабочая область пустая, так как никакая информации о магазине, торговых точках, товарах и сотрудниках не добавлена.

Сотруднику доступны вкладки: «Заказы», «Товары», «Сотрудники», «Клиенты», «Магазины», «Поставщики» и «О магазине». Во вкладке «Заказы» (рис. 25) администратор видит все заказы всех магазинов, может их изменять или удалять. Для удобства, ему предоставлена возможность фильтрации заказов по их статусам.

Во вкладке «Товары» (рис. 26) администратору видны все товары всех магазинов, для удобства просмотра он может фильтровать их по магазинам. Также в этой вкладке ему доступен импорт товаров в систему через файл-каталог. Это может быть удобно, так как процесс выгрузки товаров поштучно занимает много времени.

Во вкладке «Сотрудники» (рис. 27) администратору отображается список сотрудников всех магазинов, а также есть возможность редактировать

занимаемую ими должность, добавлять новых сотрудников и фильтровать сотрудников по торговым точкам, в которых они работают.

Во вкладке «Клиенты» (рис. 24) доступна информация о всех клиентах компании, то есть о всех пользователях, которые зарегистрировались в системе.

Во вкладке «Магазины» (рис. 28) администратор может создавать новые и редактировать уже существующие торговые точки.

Информация о поставщиках расположена во вкладке «Поставщики» (рис. 29).

Вкладка «О магазине» (рис. 30) необходима для отображения информации о магазине для внешнего пользователя, а также для указания типа продаваемого товара. Поле «Адресная доставка для всех» означает, что продажа товара разрешена через Интернет, и клиент, являющийся физическим лицом, может оформить доставку товара по конкретному адресу. Также в данной вкладке сотрудник может указать меру товара и единицы его измерения. Это необходимо для правильного отображения списка товаров пользователям. Так, в алкогольном магазине мерой товара будет объем тары, в которой поставляется товар, а единицами измерения товара – литры. В свою очередь, для книжного магазина такая мера товара не подойдет, тут ей будет являться, например, количество страниц в книге, а единицами измерения – страницы. Пример отображения информации, изменяемой во вкладке «О магазине» представлен на рис. 31.

Алексей Глазунов должность Администратор										
Заказы Товары Сотрудники Клиенты Магазины Поставщики О магазине										
Выход										
Список заказов										
Все Выполнить										
№	Клиент	Магазин	Адрес доставки	Товары в заказе	Комплектовщик	Курьер	Статус заказа	Дата заказа	Дата доставки	Действия
#30	yur@mail.ru	Склад	None	W1 - 1,00 л - 2 шт 212,00 руб Сумма 424,00 руб	Ольга Завьялова	Константин Малышев	Завершен	23 апреля 2018 г.	23 апреля 2018 г.	Изменить заказ Удалить
#29	phy@mail.ru	Магазин 1 "Центр"	Самовывоз	Пиво 1 - 1,00 л - 12 шт 122,00 руб Сумма 1464,00 руб	Олег Николаев	-	Завершен	22 апреля 2018 г.	23 апреля 2018 г.	Изменить заказ Удалить
#27	tx-tx@mail.ru	Склад	Самовывоз	w3 - 1,00 л - 6 шт 222,00 руб Виски 7 - 1,00 л - 1 шт 123,00 руб Сумма 1455,00 руб	Ольга Завьялова	Константин Малышев	Заказ комплектуется	19 апреля 2018 г.	22 апреля 2018 г.	Изменить заказ Удалить
#26	tx-tx@mail.ru	Склад	None	23r32 - 32,00 л - 5 шт 322,00 руб Виноя - 1,00 л - 6 шт 222,00 руб Сумма 2942,00 руб		Константин Малышев	Завершен	17 апреля 2018 г.		Изменить заказ Удалить
#25	tx-tx@mail.ru	Склад	None	23r32 - 32,00 л - 1 шт 322,00 руб Сумма 322,00 руб	Ольга Завьялова	Константин Малышев	Завершен	15 апреля 2018 г.	23 апреля 2018 г.	Изменить заказ Удалить

Рис. 25 – Интерфейс рабочей области для администратора компании

Алексей Глазунов должность Администратор			
Заказы Товары Сотрудники Клиенты Магазины Поставщики О магазине			
Выход			
Список товаров			
Все Выполнить Добавить Импорт			
Магазин	Название товара	Цена	Количество
Новый магазин 2	Виски 8	123,00	1
Новый магазин 2	Виски 9	123,00	3
Новый магазин 2	Виски 7	123,00	4
Склад	Виски 3	123,00	4
Склад	Виски 2	123,00	4
Склад	Виски 1	123,00	4
Магазин 1 "Центр"	Пиво 1	122,00	645
Склад	hoegaarden	32,00	1212
Склад	Пиво опилля	4,00	123
« 1 2 3 4 »			

Рис. 26 – Вкладка «Товары» для администратора компании

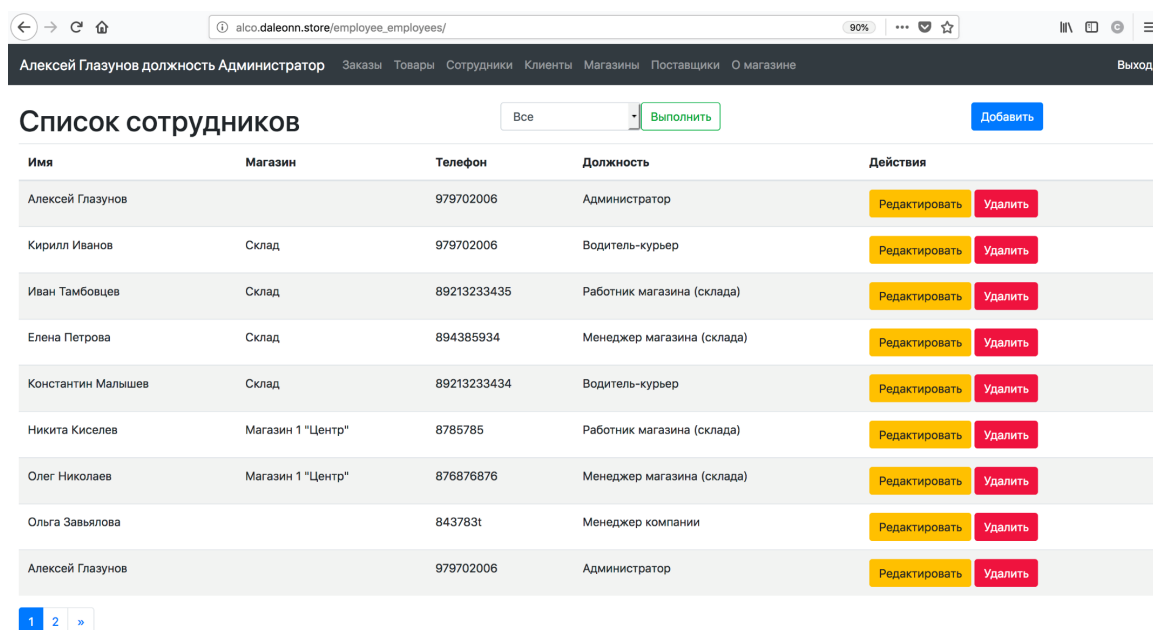


Рис. 27 – Вкладка «Сотрудники» для администратора компании

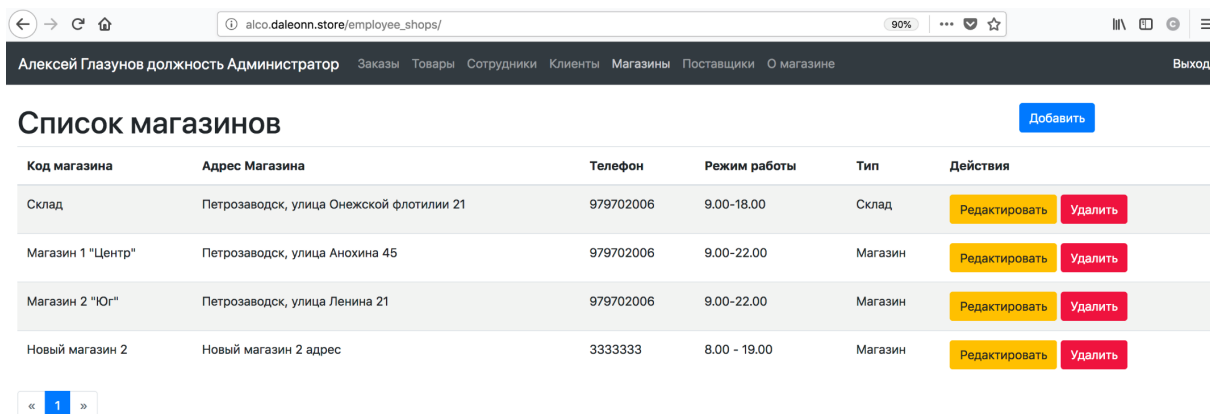


Рис. 28 – Вкладка «Магазины» для администратора компании

Данные	Телефон
Поставщик 1	979702006
Поставщик 2	979702006
Поставщик 3	979702006
Поставщик 4	None
Поставщик 5	None
Поставщик 6	None
Поставщик 7	None
Поставщик 8	None
Поставщик 9	None

Рис. 29 – Вкладка «Поставщики» для администратора компании

Добавить информацию о магазине

Название

Описание

☐ Адресная доставка для всех

Мера товара

Единицы измерения

[Добавить](#)

Рис. 30 – Вкладка «О магазине» для администратора компании

Торговый дом "Согласие"

Оптовая продажа алкоголя в Петрозаводске

Торговые точки

<p>Склад</p> <p>979702006</p> <p>9.00-18.00</p> <p>Петрозаводск, улица Онежской флотилии 21</p> <p>Перейти</p>	<p>Магазин 1 "Центр"</p> <p>979702006</p> <p>9.00-22.00</p> <p>Петрозаводск, улица Анохина 45</p> <p>Перейти</p>	<p>Магазин 2 "Юг"</p> <p>979702006</p> <p>9.00-22.00</p> <p>Петрозаводск, улица Ленина 21</p> <p>Перейти</p>	<p>Новый магазин 2</p> <p>3333333</p> <p>8.00 - 19.00</p> <p>Новый магазин 2 адрес</p> <p>Перейти</p>
--	--	--	---

Рис. 31 – Пример страницы с информацией о компании

3.5. Тестирование разработанного продукта

Для проверки корректности работы разработанной системы необходимо провести ее тестирование по различным параметрам. Во-первых, необходимо провести тестирование системы на ее корректное отображение различными устройствами и браузерами. Во-вторых, необходимо проверить корректность работы системы путем создания и комплектации интернет-магазина и имитации его работы. [23]

В наше время существует множество браузеров, которыми могут пользоваться сотрудники компаний. Чтобы не ограничивать их использованием какого-то одного браузера, необходимо провести тестирование системы на корректность отображения в самых популярных браузерах в России. Разработанный продукт рассчитан на российские компании и русскоязычных пользователей, в связи с локализацией и законами, ограничивающими продажи в Интернете в нашей стране.

В таблицах 18 и 19 представлен рейтинг браузеров, согласно занимаемой ими доли рынка в России с апреля 2017 года по апрель 2018 года. Рейтинг в таблице 18 составлен согласно данным компании Statcounter, занимающаяся сбором и анализом веб-трафика [25].

Рейтинг, место	Название браузера	Доля рынка браузеров, %
1	Google Chrome	54.73
2	Yandex Browser	12.22
3	Safari	9.56
4	Mozilla Firefox	7.76
5	Opera	7.17
6	Internet Explorer	2.6
	Иные браузеры	5.96

Таблица 18 – рейтинг браузеров в России, StatCounter [25]

Рейтинг в таблице 19 составлен согласно сервису Яндекс.Радар [26].

Рейтинг, место	Название браузера	Доля рынка браузеров, %
1	Google Chrome	40.36
2	Yandex Browser	19.81
3	Safari	10.2
4	Opera	6.91
5	Mozilla Firefox	6.61
	Иные браузеры	16.11

Таблица 19 – рейтинг браузеров в России, Яндекс.Радар [26]

Для удобства отображения данные рейтинги представлены на круговых диаграммах (рис. 32). Можно заметить, что рейтинг браузеров у двух компаний примерно одинаковый и отличие состоит только в процентной доли рынка браузеров.

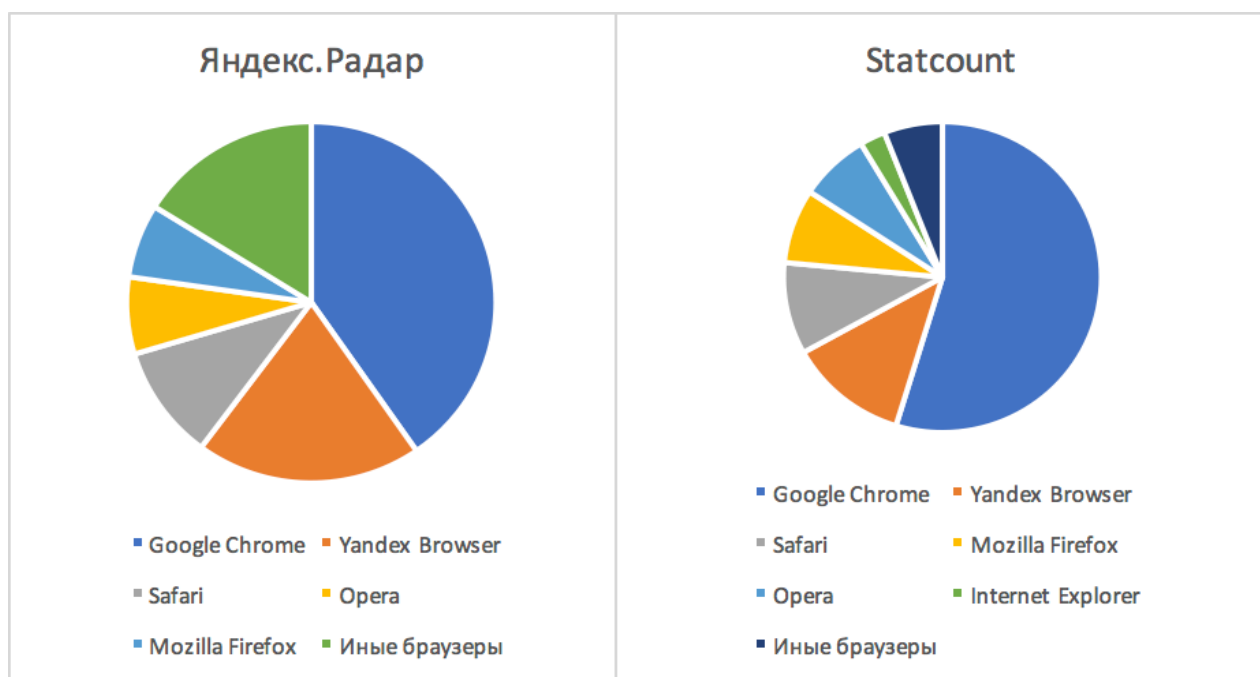


Рис. 32 – диаграммы, построенные на основании таблиц

Проведя анализ популярности браузеров, можно прийти к выводу, что самыми популярными браузерами в России на данный момент являются: Google Chrome, Yandex Browser и Safari.

Именно в этих браузерах и проводится тестирование на корректность отображения разработанного продукта. Под корректным отображением

понимается правильное отображение всех функций различных пользователей. Использование стандартизированных технологий разработки веб-интерфейса предполагает, что не должно возникнуть проблем и ошибок в корректности отображения. Процесс тестирования подтвердил это предположение. Интернет-магазины действительно корректно отображаются в самых популярных браузерах. Также, тест показал, что разработанная система корректно отображается не только в браузерах на десктопах, а также в браузерах на смартфонах и планшетах. Это позволит сотрудникам компании и ее клиентам использовать не только стационарные компьютеры и ноутбуки при работе с приложением, а также смартфоны и планшеты.

Вторым этапом тестирования является проверка созданного приложения на корректность работы. Необходимо, чтобы в процессе работы с системой не возникало различных ошибок. Этот тест был проведен путем создания интернет-магазина по продаже книг, располагающийся по адресу <http://books.daleonn.store>, и имитацией его работы. Скриншоты работы этого магазина представлены в приложении (рис. 36-38). Интернет-магазин был укомплектован товарами, магазинами и сотрудниками различных должностей. Также была симитирована работа внешнего пользователя-покупателя, то есть созданы различного вида заказы. Процесс тестирования показал, что приложение работает корректно.

Помимо этого, разработанный продукт находится в апробации у компании ООО «ТД Согласие», которая занимается оптовой продажей алкогольной продукции. Магазин этой компании находится по адресу <http://alco.daleonn.store>. Результат тестирования и отзыв о работе продукта, представлен актом о внедрении в ООО «ТД Согласие».

Подводя итог тестированию, можно сказать, что оно прошло успешно, и разработанная система работает корректно.

Выводы

В ходе анализа существующих приложений-конструкторов интернет-магазинов было выявлено, что, несмотря на их обилие и мощный функционал, не все они являются удобными в использовании. Стало понятно, что не существует универсального инструмента для создания интернет-магазина в «один клик» с удобной внутренней системой управления информационными потоками. В связи с этим, разработка нового универсального приложения-конструктора интернет-магазинов является актуальной задачей.

В ходе анализа существующих средств разработки веб-приложений были рассмотрены основные и самые популярные языки веб-программирования, такие как: PHP, Python, Ruby. Помимо этого, были проанализированы и выбраны остальные необходимые инструменты. В результате анализа выявлено, что наиболее подходящими инструментальными средствами для решения поставленной задачи являются: язык программирования Python 3.6, фреймворк языка Python Django, HTML, CSS, JavaScript, СУБД SQLite 3, веб-сервер Apache 2.0.

Проведенное тестирование и апробирование разработанного продукта путем создания двух совершенно различных интернет-магазинов:

- интернет-магазина по продаже книг, располагающийся по адресу <http://books.daleonn.store>;
- интернет-магазина по продаже алкоголя, располагающийся по адресу <http://alco.daleonn.store>;

показало его стабильную и правильную работу в обоих случаях.

Заключение

В результате выполнения данной работы было разработано актуальное сервисное веб-приложение, позволяющее создавать и комплектовать интернет-магазин, а также реализована внутренняя встроенная система по контролю и управлению заказами, товарами, сотрудниками, торговыми точками и клиентами компании. Приложение позволяет упростить эффект выхода компаний, занимающимися продажами, на рынок интернет-продаж. С его помощью можно удобно и быстро создать интернет-магазин, а также не составит труда его комплектовать.

Разработанное сервисное приложение лучше существующих аналогов, потому что выполняет тот же самый функционал, только быстрее и намного удобнее. Также имеется необходимая внутренняя встроенная система по контролю и управлению заказами, товарами, сотрудниками, торговыми точками и клиентами компании, чем не могут похвастаться конкуренты.

Разработанное сервисное веб-приложение обеспечивает полный функционал, обозначенный в постановке задачи выпускной квалификационной работы, а также отвечает всем требованиям конкурентоспособности.

С помощью созданного сервисного приложения клиенты компании могут:

- просматривать электронный каталог товаров компании, а также ее торговые точки;
- создавать заказы, состоящие из необходимых товаров, и в случае необходимости указывать адрес доставки;
- контролировать этапы выполнения заказа в личном кабинете.

Сотрудники компании, в зависимости от их должности, могут:

- создавать и описывать интернет-магазин;
- добавлять и удалять сотрудников в систему, а также изменять занимаемую ими должность;
- добавлять и изменять торговые точки в систему;
- просматривать товар, указывать те торговые точки, в которых он продается и в каком количестве, а также выгружать их в систему из файла;
- просматривать, создавать и выполнять заказы, менять их параметры;
- просматривать сведения о клиентах компании.

Список цитируемой литературы

1. Пегат А. Нечеткое моделирование и управление. М.: Бином. Лаборатория знаний, 2017. 800 с.
2. Тысленко А. Менеджмент. Организационные структуры управления. М.: Альфа-Пресс, 2011. 320 с.
3. Фролов А., Фролов Г. Создание Web-приложений: практическое руководство. М.: Русская редакция, 2001. 1040 с.
4. Экслер А. Б. Укрощение Интернета, или Самый полный и понятный самоучитель работы в Сети. М.: НТ Пресс, 2013. – 944 с.
5. Колисниченко Д. Н. PHP и MySQL. Разработка Web-приложений. СПб: БХВ-Петербург, 2015. 593 с.
6. Сафронов М. Н. Разработка веб-приложений в Yii 2. М.: ДМК-Пресс, 2015. 392 с.
7. Потапова Н. В., Большакова А. А. PHP, RUBY, PYTHON – на чем остановить свой выбор студенту при изучении веб-разработки. // Новые информационные технологии в образовании и науке: материалы X международной научно-практической конференции / 2017. С. 409–413.
8. Флэнаган Д., Мацумото Ю. Язык программирования Ruby. СПб: Питер, 2011. 496 с.
9. Лутц М. Python. Карманный справочник. М.: Вильямс, 2015. 320 с.
10. Гринберг, М. Разработка веб-приложений с использованием Flask на языке Python. М.: ДМК-Пресс, 2016. 272 с.
11. Дронов В. Django: практика создания Web-сайтов на Python. СПб.: БХВ-Петербург, 2016. 528 с.
12. Official site of Flask framework[Электронный ресурс]. – Режим доступа: <http://flask.pocoo.org/> (Дата обращения 13.01.2018г.)

13. Official site of Django framework [Электронный ресурс]. – Режим доступа: <https://www.djangoproject.com/start/overview/> (Дата обращения 13.01.2018г.)
14. Тарасов С. В. СУБД для программиста. Базы данных изнутри. Омск: Соломон, 2015. 320 с.
15. Макдональд М. Веб-разработка. Исчерпывающее руководство. СПб: Питер, 2017. 640 с.
16. Арнольд М., Алмейда Д. Д. Миллер К. Администрирование Apache. М.: Лори, 2012. 418 с.
17. Роббинс Дж. HTML5. Карманный справочник. М.: Вильямс, 2016. 192 с.
18. Мейер Э. А. CSS. Карманный справочник. М.: Вильямс, 2017. 288 с.
19. Казаков Г. С. Система автоматизации проектирования баз данных на основе модели «сущность-связь». // Информационные системы и измерительно-вычислительные комплексы: сборник докладов студентов и аспирантов кафедры «Измерительно вычислительные комплексы» на научно-технических конференциях / Под ред. В. В. Родионова, 2010. С. 25–31.
20. Рыжко А. Л., Рыбников Н. А., Рыжко Н. А. Информационные системы управления производственной компанией. М.: Юрайт, 2016. 356 с.
21. Дэйт К. Дж. Введение в системы баз данных. М.: Вильямс, 2017. 1328 с.
22. Кузнецов С. Д. Основы баз данных. М.: Бином, 2007. 488 с.
23. Вигерс К. И. Разработка требований к программному обеспечению. СПб.: БХВ-Петербург, 2018. 736 с.
24. Рейтинг популярности СУБД [Электронный ресурс]. – Режим доступа: <https://db-engines.com/en/ranking/relational+dbms/> (Дата обращения 30.04.2018г.)

25. Рейтинг популярности браузеров в РФ [Электронный ресурс]. – Режим доступа: <http://gs.statcounter.com/browser-market-share/all/russian-federation> (Дата обращения 15.05.2018г.)
26. Рейтинг популярности браузеров в РФ [Электронный ресурс]. – Режим доступа: <https://radar.metrika.yandex/browsers?tablePeriod=all&period=2017-04-01%3A2018-04-01> (Дата обращения 15.05.2018г.)

Приложение

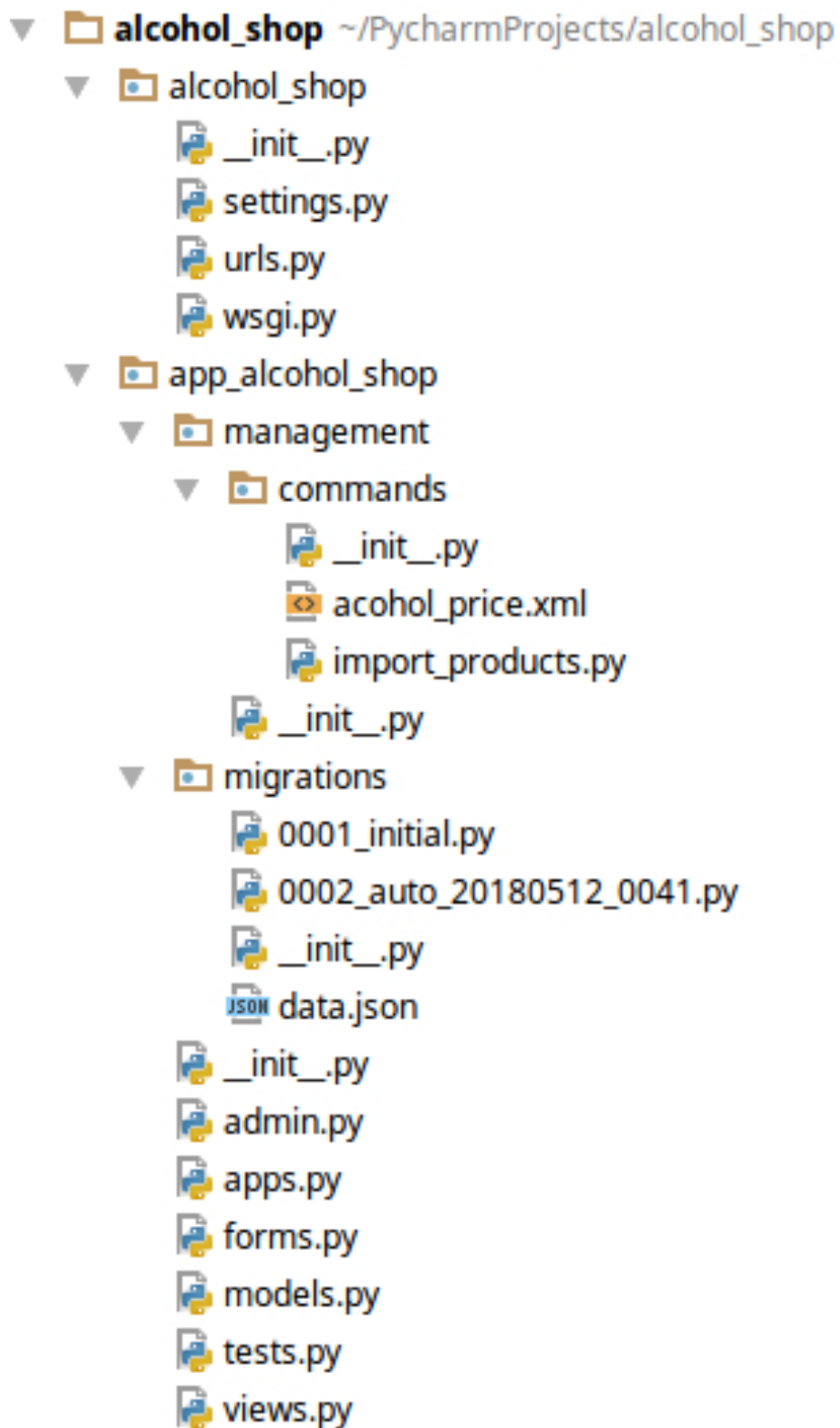


Рис. 33 – Структура директорий проекта (часть 1)

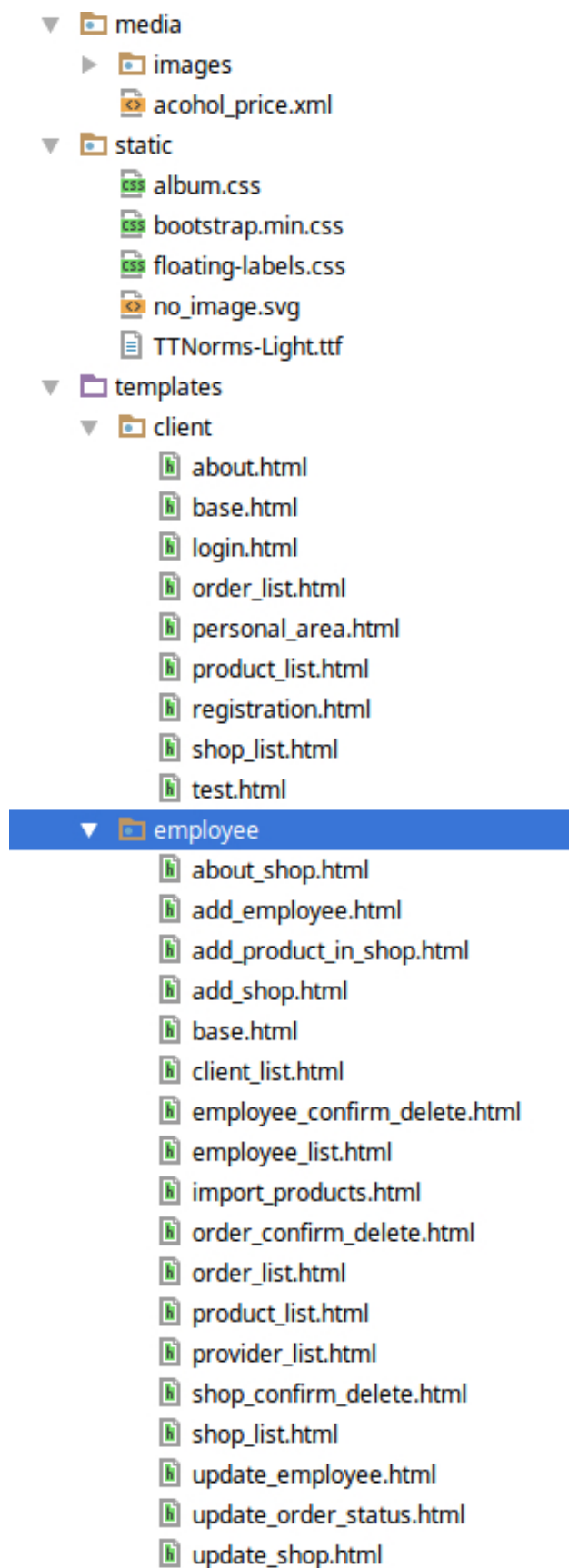


Рис. 34 – Структура директорий проекта (часть 2)

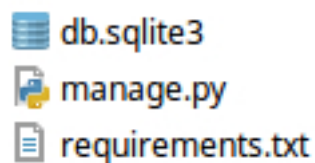


Рис. 35 – Структура директорий проекта (часть 3)

На рис. 33, 34 и 35 представлена структура директорий разработанного программного продукта, ниже представлено описание и назначение каждой директории и основных необходимых файлов.

alcohol_shop – директория проекта;

alcohol_shop – пакет проекта, который содержит следующие файлы:

__init__.py – системный файл, который дает понять интерпретатору Python что данная директория является пакетом;

settings.py – основной файл настроек проекта Django;

urls.py – основной файл, хранящий маршруты приложения, то есть описание всех url-адресов;

wsgi.py - содержит вызываемый объект application.

app_alcohol_shop – пакет приложения. В нем хранится вся реализация проекта, он содержит следующие файлы:

__init__.py – системный файл, дает понять интерпретатору Python, что данная директория является пакетом;

admin.py – файл, необходимый для добавления моделей в интерфейс администрирования;

apps.py – файл, описывающий приложение;

forms.py – файл, в котором содержатся формы для работы с данными;

models.py – файл со списком моделей;

views.py – файл с видами, которые отображают необходимые шаблоны и обрабатывают данные от пользователей.

Migrations – директория с файлами миграций базы данных;

management/commands – директория для хранения собственных команд;

import_products.py – команда для импорта товаров файлом;

Templates – директория которая хранит шаблоны, отображающиеся пользователям системы. Все шаблоны разделены на две части: которые предназначены для клиентов, (находятся в директории *client*) и которые предназначены для персонала (находятся в директории *employee*).

Шаблоны для клиентов находятся в директории *client*:

base.html – базовый файл шаблона, от которого наследуются другие шаблоны. В нем хранится общая верстка, которая используется в других шаблонах. Например, шапка сайта;

login.html – файл шаблона авторизации пользователя;

registration.html – файл шаблона регистрации пользователя;

about.html – файл шаблона, отображающий информацию о магазине;

personal_area.html – файл шаблона личного кабинета клиента;

order_list.html – файл шаблона, отображающий список заказов клиента;

shop_list.html – файл шаблона, отображающий список магазинов клиенту;

product_list.html – файл шаблона, отображающий список товаров клиенту;

Шаблоны для сотрудников находятся в директории `employee`:

base.html – базовый файл шаблона, от которого наследуются другие шаблоны. В нем хранится общая верстка, которая используется в других шаблонах. Например, шапка сайта;

about_shop.html – файл шаблона для добавления информации о магазине;

add_employee.html – файл шаблона для добавления сотрудника;

add_product_in_shop.html – файл шаблона для добавления товаров;

add_shop.html – файл шаблона для добавления магазина;

client_list.html – файл шаблона, отображающий список клиентов;

employee_confirm_delete.html – файл шаблона подтверждения удаления сотрудника;

employee_list.html – файл шаблона для отображения списка сотрудников;

import_products.html – файл шаблона для импорта товаров;

order_confirm_delete.html – файл шаблона подтверждения удаления заказа;

order_list.html – файл шаблона для отображения списка заказов;

product_list.html – файл шаблона для отображения списка товаров;

provider_list.html – файл шаблона для отображения списка поставщиков;

shop_confirm_delete.html – файл шаблона для подтверждения удаления магазина;

shop_list.html – файл шаблона для отображения списка магазинов;

update_employee.html – файл шаблона для обновления информации о сотруднике;

update_order_status.html – файл шаблона для обновления информации о заказе;

update_shop.html – файл шаблона для обновления информации о магазине.

static – директория в которой хранятся статические файлы стилей;

media – директория в которую загружаются изображения товаров;

requirements.txt – файл со списком Python пакетов, необходимых для работы с проектом;

manage.py – файл менеджер проекта;

db.sqlite3 – файл базы данных СУБД SQLite.

На рис. 36-38 представлены скриншоты созданного магазина по продаже книг.

Список товаров

Все Выполнить Добавить Импорт

Магазин	Название товара	Цена	Количество
Книжный магазин "Центр"	Компьютерные сети. 5-е изд.	942,00	20
Книжный магазин "Центр"	.NET Сетевое программирование для профессионалов	650,00	20
Книжный магазин "Центр"	Большая детская энциклопедия	250,00	20
Книжный магазин "Центр"	Динозавры. Полная энциклопедия	286,00	20
Книжный магазин "Центр"	Метелики. Міні-енциклопедія	90,00	17
Книжный магазин "Центр"	Танки и бронетехника	365,00	16
Склад	Angular 4. Быстрая разработка сверхдинамических Web-сайтов	420,00	20
Склад	XML. Базовый курс 4-е издание	1080,00	20
Склад	HTMLS + CSS3. Основы современного WEB-дизайна	320,00	20

« 1 2 3 4 »

Рис. 36 – Список товаров в книжном магазине

Список заказов

Все Выполнить

№	Клиент	Магазин	Адрес доставки	Товары в заказе	Комплектовщик	Курьер	Статус заказа	Дата заказа	Дата доставки	Действия
#9	test@mail.ru	Склад	г. Санкт-Петербург, Невский 115	Танки и бронетехника - 1,00 л - 4 шт 365,00 руб Сумма 1460,00 руб	Евгений Кудрявцев	-	Собран, готов к выдаче	24 мая 2018 г.		Изменить заказ Удалить
#8	12345678@1.ru	Книжный магазин "Юг"	Самовывоз	Метелики. Міні-енциклопедія - 1,00 л - 3 шт 90,00 руб Динозавры. Полная энциклопедия - 1,00 л - 2 шт 286,00 руб Сумма 842,00 руб		-	Заказ комплектуется	24 мая 2018 г.		Изменить заказ Удалить
#7	12345@mail.ru	Склад	Город Петергоф, университет	Метелики. Міні-енциклопедія - 1,00 л - 4 шт 90,00 руб Сумма 360,00 руб	Константин Новоселов	-	Завершен	24 мая 2018 г.	24 мая 2018 г.	Изменить заказ Удалить
#6	123@mail.ru	Книжный магазин "Юг"	Самовывоз	Метелики. Міні-енциклопедія - 1,00 л - 2 шт 90,00 руб Сумма 180,00 руб	Кирилл Игнатьев	-	Собран, готов к выдаче	17 мая 2018 г.		Изменить заказ Удалить

Рис. 37 – Интерфейс администратора книжного (поле заказов)

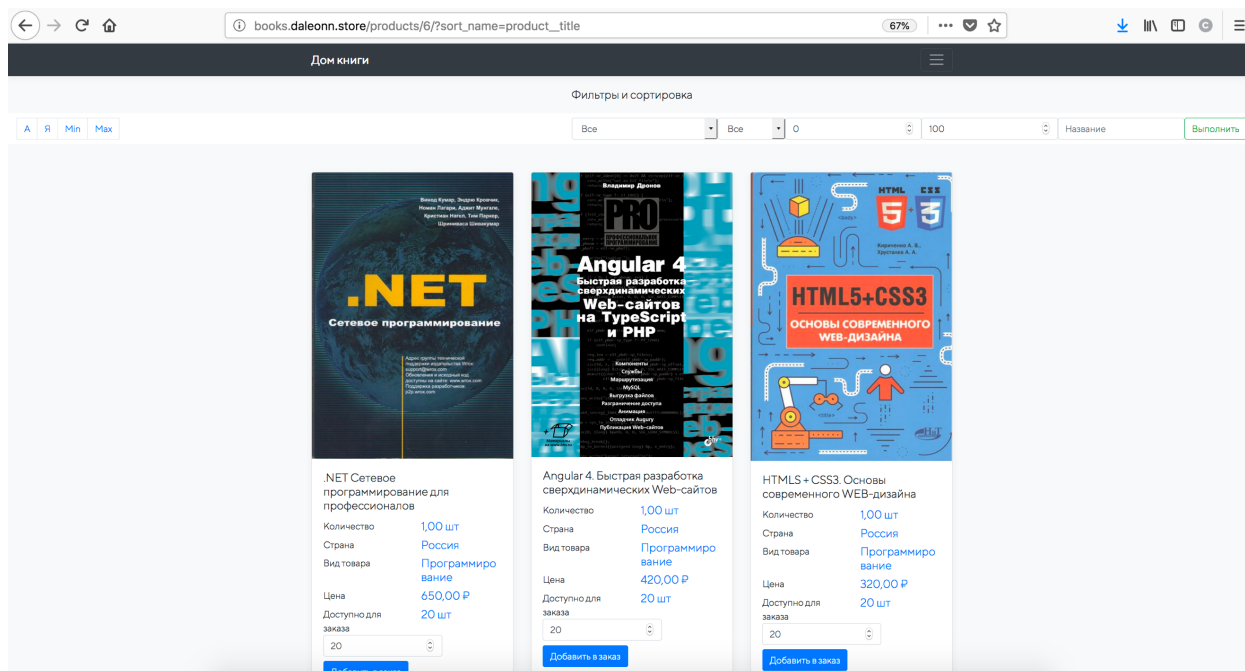


Рис. 38 – Каталог товаров книжного магазина